

POLITECNICO DI MILANO

Corso di Laurea Specialistica in Ingegneria Informatica

Dipartimento di Elettronica e Informazione



UNA METODOLOGIA DI BENCHMARKING ENERGETICO PER
SISTEMI GESTIONALI OPEN SOURCE

Relatore: Prof.ssa Chiara FRANCALANCI
Correlatore: Ing. Eugenio CAPRA

Tesi di Laurea Specialistica di:
Antonio Alberto FAMA
matr. 725481

Anno Accademico 2010/2011

*“A papá Franco, mamma Elda e Chiara,
partecipi di tutte le mie gioie e dolori.
Grazie per l’amore e il sostegno di questi anni.”*

*“Ai ragazzi dell’ICT tutti, ad Andrea e Saverio,
compagni di questo lungo viaggio,
grazie per il supporto, i consigli e l’amicizia che avete
saputo donarmi.”*

Indice

Elenco delle figure	III
Introduzione	1
1 Stato dell'arte	5
1.1 Introduzione al Green IT	5
1.1.1 L'uomo, l'energia e l'ambiente	5
1.1.2 Prospettiva energetica nell'ICT	7
1.1.3 Green IT e ambiti	16
1.2 Il Green Software	21
1.2.1 L'efficienza energetica nel Software	21
1.3 Benchmark per l'efficienza energetica	29
1.3.1 Il Benchmark TPC-Energy	29
1.3.2 Il Benchmark SPECpower_ssj2008	32
1.3.3 Limite degli Energy-Benchmark	35
2 Metodologia	37
2.1 Piano di Misura	37
2.2 Stima del consumo energetico	38
2.2.1 Modello energetico e stima indiretta del consumo	38
2.2.2 Metrica beta	41
2.2.3 Metrica gamma	46
2.3 Selezione del Software	46
2.3.1 Openbravo	48
2.3.2 Adempiere	49
2.3.3 Apache OFBiz	51
2.4 Definizione benchmark energetico	53
2.4.1 Profilazione Transazioni	53
2.4.2 Definizione Scenari	59
2.4.3 La suite HP LoadRunner	60
2.5 Strumenti di misura	65
2.5.1 Pinza Amperometrica e Multimetro Digitale	65
2.5.2 Windows Performance Monitor	67

2.5.3	CPU Stress Web Application	69
2.6	Classificazione energetica	75
3	Misure ed Analisi dei Risultati	77
3.1	Misura dell'idle	77
3.2	Stima metrica beta	79
3.2.1	OCCT Test	79
3.2.2	Prime95 Test	81
3.2.3	Web Application Test	86
3.3	Validazione Modello energetico	87
3.3.1	Test1: Inserimento nuovo Business Partner	89
3.3.2	Test2: Inserimento nuovo Prodotto	90
3.3.3	Test3: Inserimento nuovo Ordine di Acquisto	91
3.3.4	Test4: Inserimento nuovo Ordine di Vendita	92
3.3.5	Analisi dei Risultati	92
3.4	Esempio di classificazione energetica	94
4	Conclusioni e Sviluppi Futuri	95
A	Il Benchmark TPC-C	97
	Bibliografia	99

Elenco delle figure

1.1	Overview ISO26000 Responsabilità Sociale	6
1.2	Stime dei consumi energetici nell'IT. Fonte: The Green Grid	8
1.3	Stime dei consumi energetici nei paesi UE. Fonte: European Commission DG INFSO	9
1.4	Fonte: European Commission DG INFSO	10
1.5	Fonte: European Commission DG INFSO	10
1.6	Impatto dell'ICT sulle emissioni dei gas serra. Fonte: The Climate Group	11
1.7	Effetti dell'ICT divisi per settore. Fonte: The Climate Group	12
1.8	Dati sull'efficienza energetica dell'hardware	14
1.9	Spesa mondiale per i server (miliardi di dollari. Fonte: IDC (2006))	15
1.10	Rapporto spesa per energia e raffreddamento/spese per l'ac- quisto di nuovi server (Percentuale). Fonte: IDC (2006)) . . .	15
1.11	Ripartizione del consumo energetico di un data center	17
1.12	Energia assorbita da un data center (Watt, Source: Renzi, 2007)	17
1.13	Efficienza energetica stimata rispetto all'efficienza massima teorica degli attuali sistemi IT suddivisi per diversi livelli logici/strutturali	18
1.14	Modello ISO/IEC 9126	22
1.15	Potenza consumata nei sistemi ERP	24
1.16	Potenza consumata nei sistemi CRM	25
1.17	Potenza consumata nei DBMS	25
1.18	Metriche TPC	30
1.19	Reported Energy Configuration	31
1.20	Power Measurable Unit	31
1.21	EMS	32
1.22	Architettura del benchmark SPECpower_ssj2008. Fonte: SPE- Cpower_ssj2008 User Guide	33
1.23	Consumo energetico a diversi carichi.	34
2.1	Paragone automobilistico	40
2.2	Esempi di caratteristica del processore	42

2.3	Interfaccia di OCCT	43
2.4	Grafici generati da OCCT al termine di un test	44
2.5	Interfaccia di Prime95	45
2.6	Metodo dei Trapezi	46
2.7	Openbravo Platform	49
2.8	Openbravo screenshot	50
2.9	Adempiere screenshot	51
2.10	OFBiz screenshot	52
2.11	Processo di Testing	61
2.12	Planning del Processo di Test	62
2.13	Esempio di architettura del SUT	62
2.14	Pinza Amperometrica	66
2.15	Multimetro digitale Extech 383274	66
2.16	Performance Monitor di Windows Server 2008	67
2.17	Performance Monitor: creazione del Set Agent	68
2.18	Processo di eProcurement	69
2.19	Tabella Utenti creata per il controllo d'accesso	70
2.20	Tabella Ordine per la gestione delle informazioni di un ordine	70
2.21	Ciclo di Vita dell'Ordine	71
2.22	Pagina iniziale di autenticazione	72
2.23	Homepage	72
2.24	Creazione nuova RdA	73
2.25	L'RdA creata viene inserita nell'elenco RdA, da cui é possibile creare l'ordine oppure cancellare il record corrispondente. . .	73
2.26	L'ordine creato viene inserito nell'elenco OdA pendenti da cui é possibile effettuare il pagamento e, solo successivamente, l'archiviazione. Non é dunque possibile archiviare un ordine prima che venga estinto il pagamento	74
2.27	Ordine pagato e pronto per essere archiviato	74
2.28	Archiviazione dell'ordine	74
3.1	Misura dell'idle	78
3.2	Risultati delle misure in idle	78
3.3	OCCT Test	79
3.4	Punti stimati con OCCT	80
3.5	Caratteristica ottenuta con OCCT	80
3.6	Prime95 - 1 core	81
3.7	Prime95 - 2 core	81
3.8	Prime95 - 3 core	82
3.9	Prime95 - 4 core	82
3.10	Prime95 - 5 core	83
3.11	Prime95 - 6 core	83
3.12	Prime95 - 7 core	84
3.13	Prime95 - 8 core	84

3.14	Punti stimati con Prime95	85
3.15	Caratteristica ottenuta con Prime95	85
3.16	Potenza ed utilizzo CPU tramite la Web Application	86
3.17	Web Application - punti candidati	86
3.18	Caratteristica identificata tramite la Web Application	87
3.19	Potenza - Nuovo Business Partner	89
3.20	Utilizzo - Nuovo Business Partner	89
3.21	Risultati - Nuovo Business Partner	89
3.22	Potenza - Nuovo Prodotto	90
3.23	Utilizzo - Nuovo Prodotto	90
3.24	Risultati - Nuovo Prodotto	90
3.25	Potenza - Nuovo Ordine di Acquisto	91
3.26	Utilizzo - Nuovo Ordine di Acquisto	91
3.27	Risultati - Nuovo Ordine di Acquisto	91
3.28	Potenza - Nuovo Ordine di Vendita	92
3.29	Utilizzo - Nuovo Ordine di Vendita	92
3.30	Risultati - Nuovo Ordine di Vendita	92
3.31	Tabella riassuntiva metriche gamma	94
3.32	Parametri statistici derivati	94
3.33	Esempio di classificazione energetica per i tre sistemi ERP	94
A.1	Modello dei dati della specifica TPC-C	98
A.2	Distribuzione frequenza della transazioni	99

Introduzione

Con il crescere del nostro fabbisogno energetico degli ultimi decenni, é aumentata anche la nostra dipendenza dai combustibili fossili (petrolio, gas naturale e carbone), che rilasciano nell'atmosfera ingenti volumi di biossido di carbonio, ossido di azoto e metano.

I combustibili fossili rappresentano attualmente circa l'80% del consumo di energia. Tali emissioni sono responsabili non solo dei mutamenti del clima, ma anche dell'innalzamento dei livelli di smog e di fenomeni quali le piogge acide, rappresentando una crescente minaccia per la salute dell'uomo.

Per questi motivi, temi quali sviluppo sostenibile ed efficienza energetica hanno assunto negli ultimi decenni una notevole importanza non solo nell'industria e nelle comunità scientifiche, ma anche tra i governi dei diversi paesi e nelle istituzioni internazionali, attraverso l'adozione e la definizione di norme volte a responsabilizzare gli individui e le organizzazioni sull'impatto delle loro attività su società ed ambiente.

Le tecnologie ICT, ovvero l'insieme di tutti i dispositivi e servizi atti alla ricezione, elaborazione e trasmissione elettronica dell'informazione, giocano un duplice ruolo sull'impatto ambientale. Da un lato l'introduzione di moderne tecnologie ha comportato un incremento dei consumi energetici, dall'altro l'innovazione ICT propone soluzioni efficienti dal punto di vista dell'impatto ambientale e dello sviluppo sostenibile. L'assenza in passato di una prospettiva energetica dell'ICT é imputabile agli enormi benefici che l'IT stessa ha portato nelle organizzazioni, comportando un parziale disinteresse verso aspetti legati alla tutela dell'ambiente.

La maggior parte dei manager dell'IT sembra credere che il software di per sé abbia un impatto molto limitato sui consumi, specialmente nell'ambito dei sistemi enterprise. É generalmente accettato che i sistemi operativi influenzino le prestazioni e i consumi di energia, ma non le applicazioni. Queste convinzioni, tuttavia, raramente sono supportate da un approccio scientifico e da misure attendibili.

In realtà il software rappresenta il primo vero e proprio responsabile del consumo energetico in quanto, a seconda di come sfrutta e utilizza le risorse a sua disposizione, può determinare un aumento o una diminuzione dell'assorbimento di energia.

Misurare direttamente l'energia utilizzata da porzioni di codice e utilizzare tale metrica per analizzare l'efficienza del software vorrebbe dire comunque dipendere dall'hardware su cui il programma, la transazione o il codice viene fatto eseguire, in quanto l'elaborazione richiede energia in modo dipendente dall'efficienza dell'hardware. Se invece pensassimo di prendere come metriche di riferimento solamente grandezze legate all'utilizzo delle risorse da parte di un software, e non l'energia direttamente consumata, potremmo svincolarci in parte dalla dipendenza dell'hardware nella classificazione.

Differentemente dal mondo hardware, non si hanno ad oggi tecniche e metodologie consolidate di etichettatura del software dal punto di vista energetico, poiché non si hanno a disposizione analisi, benchmark di riferimento e misurazioni su larga scala del consumo derivante dal software applicativo. Il limite di recenti soluzioni di benchmarking per l'efficienza energetica, quali SPECpower ssj2008 o la specifica TPC-Energy, risiede nel fatto che sono finalizzate al testing di un intero sistema piuttosto che dell'applicazione finale.

La difficoltà nella definizione delle metriche di efficienza energetica del software risiede nella quantificazione del lavoro svolto da un'applicazione con una quantità unitaria di energia. Ciò è vero soprattutto nei sistemi ERP, in cui occorre considerare differenti tipologie di transazione e quindi diverse metriche di efficienza.

Il lavoro di questa tesi si inserisce nel filone di ricerca sul Green Software condotto presso il Dipartimento di Elettronica ed Informazione del Politecnico di Milano e prosegue lo studio relativo all'identificazione di metriche per la misurazione dell'efficienza energetica del software.

L'obiettivo è la validazione di un modello per la stima indiretta del consumo energetico di sistemi ERP open source attraverso la definizione di un benchmark specifico per la classe dei sistemi ERP. La progettazione del benchmark passa attraverso l'identificazione di workload tipici, la definizione di un insieme di transazioni standard e termina con una proposta metodologica di classificazione energetica del software.

La presentazione del lavoro è così strutturata:

capitolo 1 : viene dapprima introdotto il concetto di prospettiva energetica dell'ICT come risposta all'aumento delle emissioni dei gas serra causati dalla crescente diffusione delle tecnologie IT. Viene poi introdotto e discusso il tema del green software e tre diverse declinazioni o ambiti di studio contestuali: l'efficienza energetica dell'IT, la gestione eco-compatibile del ciclo di vita dell'IT e l'utilizzo dell'IT come strumento per una green governance. Il capitolo termina con la descrizione di due soluzioni di benchmarking per l'efficienza energetica dei sistemi

IT attualmente presenti sul mercato.

capitolo 2 : vengono definite le metodologie di misurazione indiretta del consumo energetico e di benchmarking del software, a partire dall'analisi del consumo energetico e di risorse computazionali di alcune transazioni di tre sistemi ERP open source.

capitolo 3 : vengono esposti i risultati di validazione della metodologia di misura illustrata nel capitolo 2, seguiti da un esempio pratico della metodologia di classificazione energetica del software proposta, effettuata sulla classe dei sistemi ERP analizzati.

Capitolo 1

Stato dell'arte

1.1 Introduzione al Green IT

I costi energetici sono drammaticamente aumentati negli ultimi anni e il loro impatto sui costi operativi dei data center è diventato sempre più significativo. Secondo l'IDC ¹ i costi energetici costituiscono più del 30% dei costi operativi di un data center. I consumi energetici rappresentano inoltre un limite alla scalabilità dei sistemi, in quanto assorbono potenze sempre maggiori [33]. Investire nel green IT comporta, come si vedrà in seguito, non solo un miglioramento dell'impatto ambientale, ma anche soluzioni economicamente efficienti.

1.1.1 L'uomo, l'energia e l'ambiente

La riduzione dei mutamenti climatici del nostro pianeta rappresenta una delle più grandi sfide del XXI secolo che l'uomo si trovi ad affrontare. Secondo recenti studi, condotti da esperti climatologi, il livello di emissione dei gas serra, che costituisce la principale causa di fenomeni quali innalzamento della temperatura e aumento di carbonio, è cresciuto oltre ogni predizione. Il riscaldamento globale è causato dall'eccessiva produzione e dallo smodato consumo di energia elettrica da parte dell'uomo. Con il crescere del nostro fabbisogno energetico, aumenta anche la nostra dipendenza dai combustibili fossili (petrolio, gas naturale e carbone), che rilasciano nell'atmosfera ingenti volumi di CO_2 ², N_2O ³ e CH_4 ⁴. I combustibili fossili rappresentano attualmente circa l'80% del consumo di energia. Tali emissioni sono responsabili non solo dei mutamenti del clima, ma anche dell'innalzamento dei livelli di smog e di fenomeni quali le piogge acide, rappresentando una crescente minaccia per la salute dell'uomo.

¹International Data Corporation

²biossido di carbonio

³ossido di azoto

⁴metano

Per questi motivi, temi quali sviluppo sostenibile ed efficienza energetica hanno assunto negli ultimi decenni una notevole importanza non solo nell'industria e nelle comunità scientifiche, ma anche tra i governi dei diversi paesi e nelle istituzioni internazionali, attraverso l'adozione e la definizione di norme volte a responsabilizzare gli individui e le organizzazioni sull'impatto delle loro attività su società e ambiente [2]. Si pensi alla norma ISO 26000 "Guida sulla responsabilità sociale" approvata nel novembre 2010, che intende fornire una guida mirata alla creazione di un'etica consistente con gli interessi della società e dello sviluppo sostenibile [32].



Figura 1.1: Overview ISO26000 Responsabilità Sociale

Nonostante negli anni passati l'Unione Europea abbia ricoperto un ruolo di primo piano nell'elaborazione dei due trattati internazionali sui cambiamenti climatici (la Convenzione Quadro delle Nazioni Unite del 1992 e il relativo Protocollo di Kyoto approvato nel 1997), successive prove scientifiche hanno mostrato come i risultati ottenuti siano insufficienti e si renda necessaria un'azione globale molto più ambiziosa per evitare che l'effetto serra raggiunga livelli nefasti. Nel dicembre 2008 l'Unione Europea ha adottato una strategia integrata in materia di energia e cambiamenti climatici, con la speranza di indirizzare l'Europa verso uno futuro sostenibile attraverso lo sviluppo di un'economia a basse emissioni di CO_2 ed improntata all'efficienza energetica. La strategia prevede 3 obiettivi:

- ridurre l'emissione di gas serra entro il 2020 del 20% rispetto ai livelli del 1990
- ridurre i consumi energetici del 20% attraverso un aumento dell'efficienza energetica
- soddisfare il 20% del nostro fabbisogno energetico mediante l'utilizzo delle energie rinnovabili

Alla conferenza ONU di dicembre 2009 sui cambiamenti climatici, l'UE ha dato il suo sostegno al cosiddetto accordo di Copenaghen, considerato il primo passo verso un trattato globale giuridicamente vincolante che sostituisca il protocollo di Kyoto nel 2013. L'accordo riconosce che il surriscaldamento del pianeta non deve superare i 2°C, se si vogliono evitare conseguenze catastrofiche. Per rimanere al di sotto di questa soglia, le emissioni globali devono stabilizzarsi prima del 2020, per poi essere almeno dimezzate rispetto ai livelli del 1990 entro il 2050, continuando a diminuire anche in seguito [1]. La riduzione dei consumi energetici che, come abbiamo visto è la prima causa del surriscaldamento globale, rappresenta dunque la chiave per l'abbassamento delle emissioni dei gas serra e del loro impatto sull'ambiente.

1.1.2 Prospettiva energetica nell'ICT

Nel paragrafo precedente si è parlato di sviluppo sostenibile, definito come il tentativo di rispondere ai crescenti bisogni e creare opportunità sociali senza compromettere le opportunità delle generazioni future [28]. Oggi, sviluppo e innovazione tecnologica sono due concetti strettamente correlati. Nel corso degli anni l'uso della tecnologia è cresciuto enormemente in tutti i settori, migliorando notevolmente la qualità della nostra vita [21]. In particolare nel settore IT, la diffusione di tecnologie ICT degli ultimi anni, ha rivoluzionato completamente il nostro modo di comunicare. Servizi come Google, iTunes, Twitter, Facebook sono solo alcuni esempi che testimoniano il rapido progresso nell'era della digitalizzazione. Attraverso il Web, si stima complessivamente uno scambio di circa 1.2 zettabytes ⁵ dati (si pensi ad esempio all'invio di mail, al posting di messaggi su Facebook e Twitter, al download di file musicali da iTunes ...) e un incremento della digitalizzazione di un fattore 44 entro il 2023. Nel 2011 la spesa mondiale in comunicazione ha raggiunto i 4.34 trilioni di dollari e si prevede di raggiungere i 5 entro il 2013 [26].

Secondo un'indagine condotta da ITU ⁶, il numero di utenti internet nel mondo è raddoppiato nel quinquennio 2005-2010 arrivando a toccare quota 2 miliardi di utenti connessi. Si calcola che entro il 2025 l'informazione

⁵1 zettabyte = 10^{21} byte

⁶International Telecommunication Union

circolante attraverso sistemi informatici sarà 100-200 volte il valore attuale. [1].

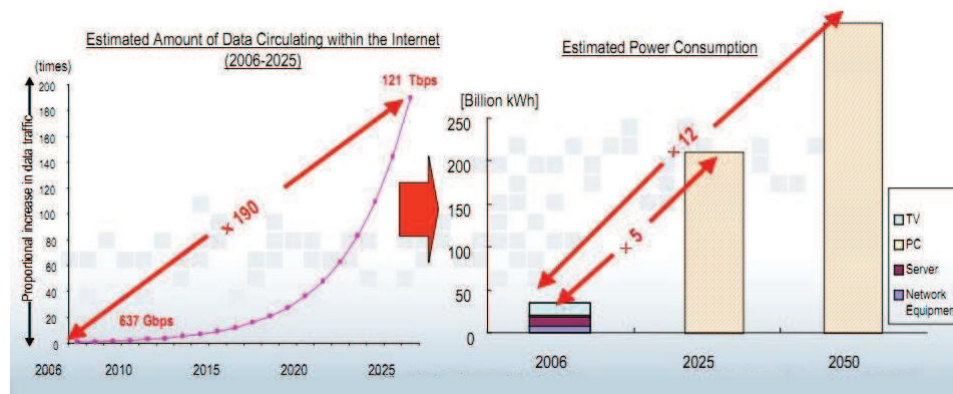


Figura 1.2: Stime dei consumi energetici nell'IT. Fonte: The Green Grid

Come si può osservare in figura 1.2, accanto all'esplosione di Internet come mezzo di comunicazione si accompagnano, non solo un aumento del numero di dispositivi di elaborazione sempre più complessi, ma anche consumi energetici sempre maggiori.

Le tecnologie ICT, ovvero l'insieme di tutti i dispositivi e servizi atti alla ricezione, elaborazione e trasmissione elettronica dell'informazione, giocano un duplice ruolo sull'impatto ambientale. Se da un lato l'introduzione di moderne tecnologie ha comportato un incremento dei consumi energetici, dall'altro l'innovazione ICT propone soluzioni efficienti dal punto di vista dell'impatto ambientale e dello sviluppo sostenibile. Esiste quindi un trade-off, di difficile valutazione, tra i benefici che l'introduzione di tecnologie green comporta e i problemi derivanti da un crescente consumo di energia elettrica. Huang [25], sostiene che in passato l'assenza di una prospettiva energetica dell'ICT è imputabile agli enormi benefici che l'IT stessa ha portato nelle organizzazioni, comportando un parziale disinteresse verso aspetti legati alla tutela dell'ambiente. Oggi, invece, la crescita esponenziale dei consumi energetici sta catturando sempre più l'attenzione della comunità scientifica, dei produttori di tecnologia e dei responsabili dei sistemi informativi.

In uno studio recentemente condotto all'università di Francoforte [2], si propongono tre fattori che spingono le aziende verso lo sviluppo sostenibile:

- fattore economico: la riduzione dei costi energetici oggi rappresenta un vantaggio competitivo sul mercato
- fattore politico: i governi incentivano l'adozione di comportamenti eco-sostenibili attraverso leggi e regolamentazioni
- fattore sociale : il mercato è sempre più attento e sensibile a soluzioni a basso impatto ambientale.

Per quanto riguarda l'impatto ambientale dell'IT, recenti studi [23, 24], affermano che il settore IT è responsabile di più del 2% delle emissioni di CO_2 e si stima una crescita da 0.53 Gt CO_2e ⁷ nel 2002 a 1.43 Gt CO_2e entro il 2020. L'EPA⁸, l'agenzia americana fondata nel 1970 per la tutela dell'ambiente e della salute dell'uomo, afferma che il consumo energetico dei data center ammonta a circa l'1.5% del fabbisogno energetico degli Stati Uniti e tale stima è in costante aumento [2]. I dati sopra citati sono confermati anche a livello europeo, come mostra la tabella 1.3: Nel 2005 il fabbisogno

	2005	2020 BAU	2020 ECO
Total ICT sector electricity use in EU 25 (TWh/a)	214.5	409.7	288.2
ICT sector without consumer electronics in EU-25 (TWh/a)	118.6	245.1	185.2
Total ICT sector electricity use in EU-27 (TWh/a)	216.0	433.1	304.7
ICT sector without consumer electronics in EU-27 (TWh/a)	119.4	259.1	195.8
Share of the ICT sector electricity use over total EU-27 electricity use (%)	7.8%	10.9%	7.7%
Share of the ICT sector electricity use (without consumer electronics) over total EU-27 electricity use (%)	4.3%	6.5%	4.9%

Figura 1.3: Stime dei consumi energetici nei paesi UE. Fonte: European Commission DG INFSO

energetico europeo relativo al settore IT era circa l'8% del consumo energetico complessivo. La tabella mostra le proiezioni per il 2020 in due scenari che si differenziano in base ai potenziali benefici derivanti dall'utilizzo di soluzioni ICT efficienti. In uno scenario BAU⁹, che considera un livello di coinvolgimento dell'ICT allineato con i trend correnti, si stima un incremento del fabbisogno energetico pari a circa l'11%, percentuale che si mantiene invece pressochè costante in uno scenario ECO, in cui si assume un coinvolgimento forte di soluzioni ICT a basso impatto.

I grafici in figura 1.4 e 1.5 mostrano l'andamento stimato dei consumi energetici dell'IT dal 2005 al 2020 nei due scenari di riferimento. Da questi dati emerge chiaramente come investire in soluzioni ICT green sia l'unica strada per controllare le emissioni a fronte di una domanda energetica in continua crescita [1].

Nel 2008 l'organizzazione no-profit The Climate Group, in collaborazione con la Global e-Sustainability Initiative (GeSi), ha pubblicato un report

⁷1 Gt CO_2e = 1 miliardo di tonnellate di CO_2 equivalente

⁸United States Environmental Protection Agency

⁹Business as usual

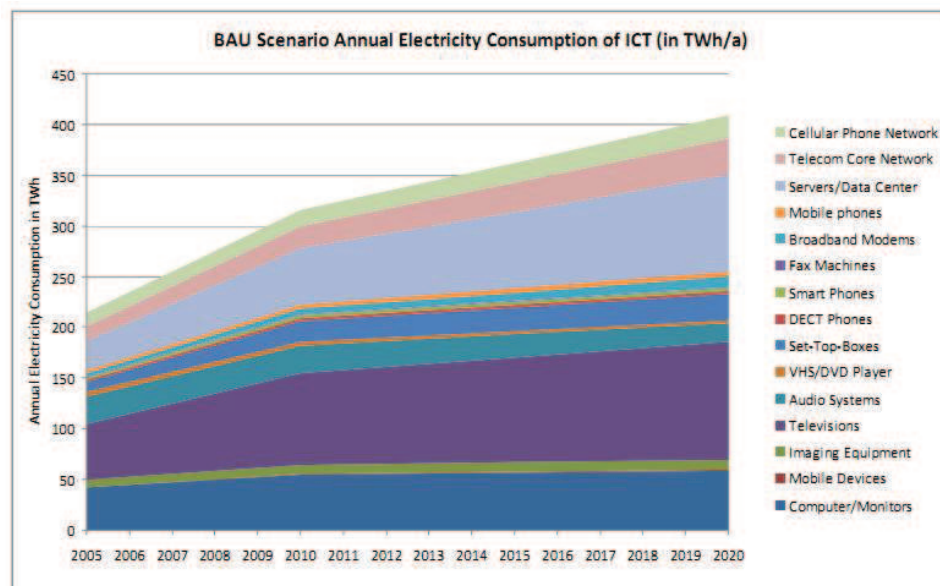


Figura 1.4: Fonte: European Commission DG INFSO

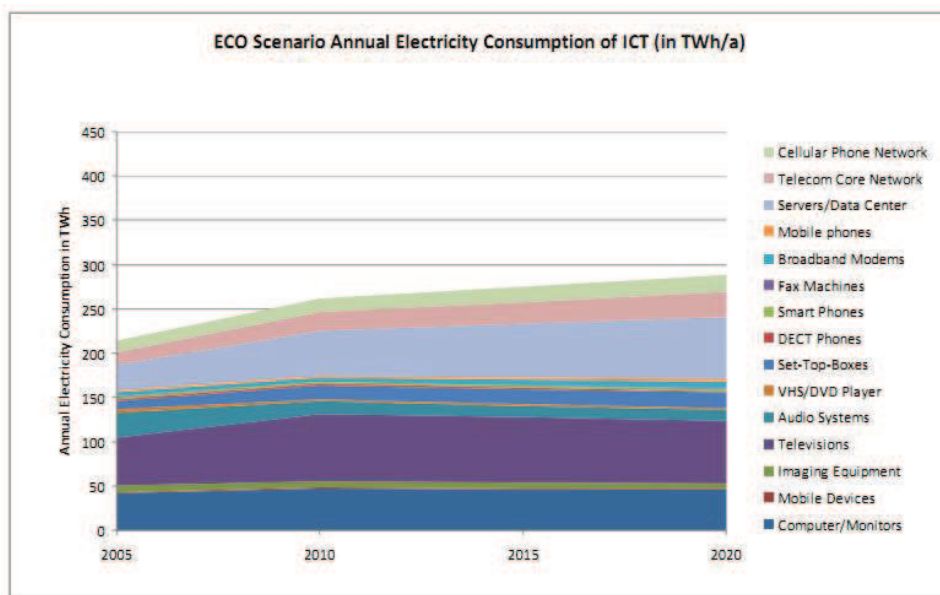


Figura 1.5: Fonte: European Commission DG INFSO

[24] in cui si esamina l'impatto delle tecnologie ICT sul surriscaldamento globale. Secondo l'indagine condotta, l'ICT gioca un ruolo fondamentale non solo nella riduzione delle emissioni ma anche nell'abbattimento dei costi energetici. Il grafico in figura 1.6 evidenzia chiaramente l'impatto positivo dell'ICT in uno scenario BAU.

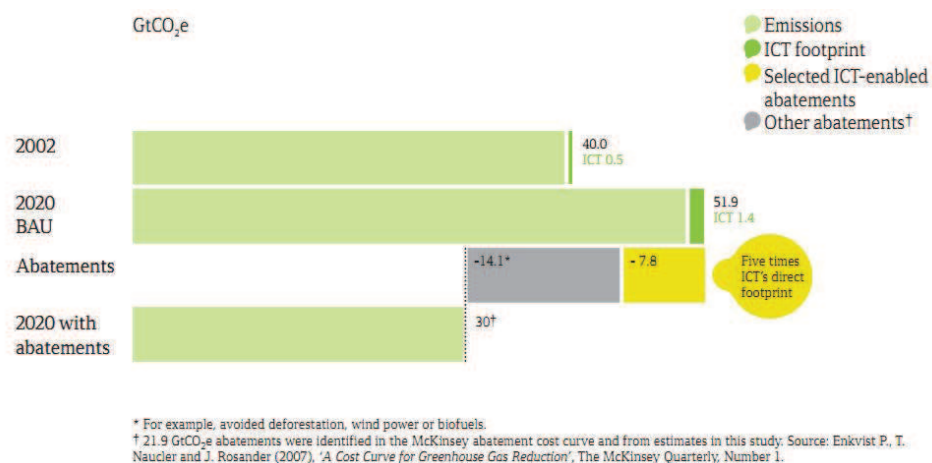


Figura 1.6: Impatto dell'ICT sulle emissioni dei gas serra. Fonte: The Climate Group

Entro il 2020, si stima che l'ICT sarà responsabile di circa il 2% delle emissioni di gas serra. Di contro, l'utilizzo di tali tecnologie consentirà un abbattimento delle emissioni di circa il 15% (circa 7.8 GtCO₂e), pari a 5 volte il suo impatto. Tali abatimenti si traducono in un risparmio dei costi energetici pari a 600 miliardi di euro.

Il diagramma in figura 1.7 mostra invece i settori più promettenti in cui l'ICT può risultare efficace nell'abbattimento delle emissioni dei gas serra. Come si nota, l'ICT contribuisce alla riduzione delle emissioni in maniera decisa nei settori dei trasporti (2.2 GtCO₂e), delle infrastrutture (2.4 GtCO₂e) e nel settore energetico (2.1 GtCO₂e).

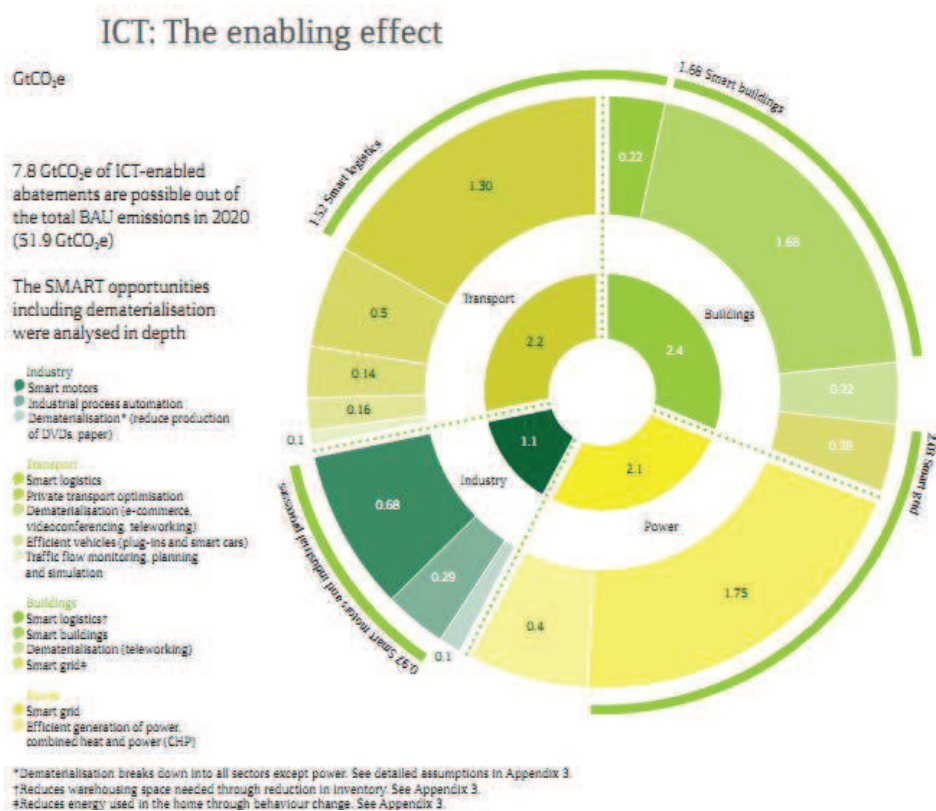


Figura 1.7: Effetti dell'ICT divisi per settore. Fonte: The Climate Group

L'infrastruttura IT di un'azienda è, negli ultimi anni, cresciuta profondamente da un punto di vista della complessità. L'avvento di processori sempre più veloci, storage ad elevata capacità e architetture hardware a basso costo ed elevata densità ha portato a data center certamente più performanti, mettendo in evidenza una serie di problematiche quali:

- aumento dei costi operazionali
- aumento del consumo energetico
- difficoltà nel rendere l'infrastruttura scalabile

Storicamente, l'acquisto o la realizzazione di un data center si è sempre basata su parametri prestazionali o di costo per prestazione, trascurando aspetti legati ai consumi energetici. L'evoluzione tecnologica degli ultimi anni ha comportato lo sviluppo di sistemi di elaborazione sempre più complessi e sempre più esigenti dal punto di vista del consumo di energia elettrica. Un data center di medie dimensioni può consumare anche 300 kW, l'equivalente di 100 appartamenti, mentre un data center di una grande banca o di un operatore Telecom può arrivare a consumare diversi MW.

L'efficienza energetica dei data center rappresenta un tema di ricerca ed innovazione su cui aziende e centri di ricerca stanno ormai lavorando da anni. Un indicatore comunemente utilizzato per misurare l'efficienza energetica di un data center è il PUE¹⁰, calcolato come la potenza entrante in un data center divisa per la potenza effettivamente assorbita dal carico IT. Un valore di PUE pari a 1 indica che tutta l'energia assorbita dall'impianto viene utilizzata per gli apparati IT. In base alle valutazioni di Green Grid (il consorzio che ha definito il PUE) e dell'agenzia EPA¹¹, il valore medio attuale del PUE per i data center sparsi nel mondo si aggira intorno a 1,8. Oggi, i data center più efficienti al mondo sono il data center "Merlin" di Capgemini, situato vicino a Londra, che raggiunge un PUE pari a 1,1 e il data center di Yahoo, costruito nelle vicinanze delle Cascate del Niagara (in condizioni climatiche particolarmente favorevoli), che presenta un PUE pari a 1,08.

I metodi per abbassare il PUE e quindi migliorare l'efficienza energetica dei data center consistono per esempio nell'uso di apparecchiature efficienti sul piano energetico, nel miglioramento dell'areazione per ridurre le esigenze di raffreddamento, nella progettazione attenta all'ambiente e investendo in software di gestione energetica. Le case produttrici di hardware svolgono da anni ricerche in questo senso e, in effetti, l'efficienza energetica dell'IT è molto aumentata, in parte spinta dalla diffusione dei dispositivi mobili e dalla necessità di aumentare l'autonomia delle batterie. Come si può osservare dai dati riportati in figura 1.8, considerando tre diverse generazioni di

¹⁰Power Usage Effectiveness

¹¹Environmental Protection Agency

processori, mentre la frequenza di clock è aumentata di un fattore 72, la potenza assorbita è aumentata solamente di un fattore 12, con un conseguente aumento dell'efficienza di un fattore 6.

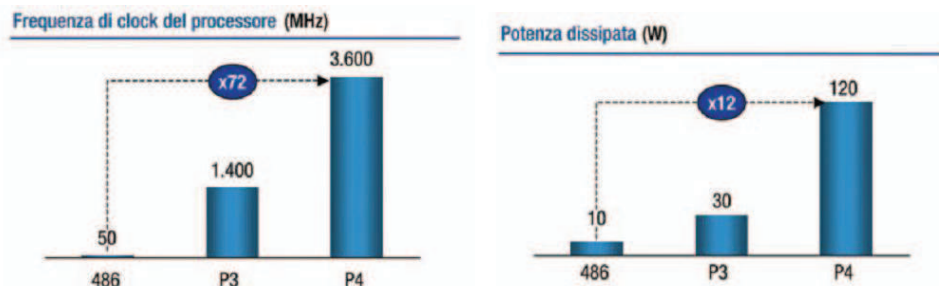


Figura 1.8: Dati sull'efficienza energetica dell'hardware

Una seconda leva efficace è poi la virtualizzazione, che è la prima tecnologia che ha consentito un approccio al Green ICT centrato sul software. L'idea è semplice: un software host (un programma di controllo) crea un ambiente simulato (la virtual machine) per il suo software ospite. Il software ospite, che può essere a sua volta un sistema operativo completo, gira come se fosse installato su una piattaforma hardware stand-alone. In questo modo ogni server fisico ospita server virtuali multipli e ciò consente ai data center di consolidare la loro infrastruttura, usando così meno elettricità e semplificando i data center stessi. La virtualizzazione dei server prevede un'applicazione software che divide un server fisico in più ambienti virtuali. Un miglior utilizzo dei server riduce le necessità di spazio fisico dove allocarli e riduce la richiesta di energia di un data center [15].

La realizzazione, l'utilizzo e la manutenzione di un sistema IT costituiscono aspetti critici che il management IT non può sottovalutare. Con la crisi economica mondiale degli ultimi anni, i CIO¹² delle aziende dispongono di budgets sempre più ristretti e, limitare la spesa informatica garantendo un buon livello di support al business, rappresenta una vera e propria sfida. I costi energetici dell'IT, in passato trascurati o annegati nei consumi energetici dell'intera azienda, costituiscono oggi una voce di costo di primaria importanza che i responsabili IT non possono più trascurare. Tali costi, infatti, coprono una parte significativa del TCO¹³ e sono in continua crescita [23]. La figura 1.9 mostra i dati relativi alla spesa mondiale per i server negli ultimi anni.

Come si osserva, mentre il costo di acquisto dell'hardware è pressoché invariato, i costi di alimentazione e di raffreddamento sono quadruplicati in poco più di un decennio. Oggi i costi energetici e di raffreddamento

¹²Chief Information Officer

¹³Total Cost of Ownership

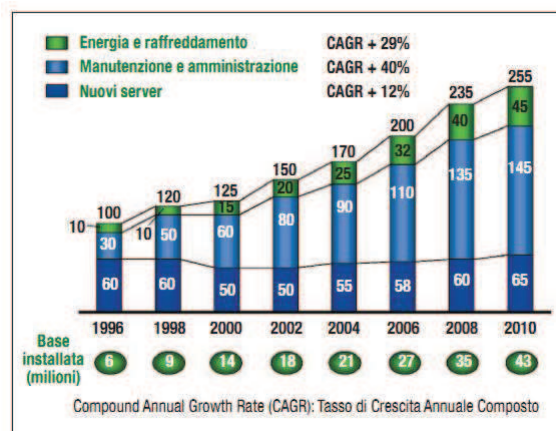


Figura 1.9: Spesa mondiale per i server (miliardi di dollari. Fonte: IDC (2006))

costituiscono circa il 60% della spesa in nuove infrastrutture, con un impatto più che significativo sul TCO (figura 1.10).

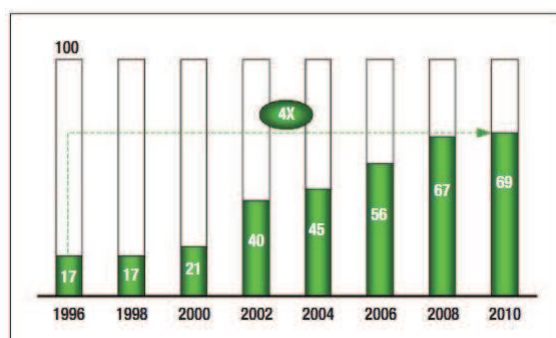


Figura 1.10: Rapporto spesa per energia e raffreddamento/spese per l'acquisto di nuovi server (Percentuale). Fonte: IDC (2006))

Un'ultima considerazione riguarda la scalabilità dei data center. Il crescente consumo di energia sta rappresentando sempre più un limite alla scalabilità, soprattutto quelli dislocati in aree ad alta densità abitativa. La potenza richiesta cresce dell'8-10% all'anno e i gestori della rete elettrica nazionale rischiano di non essere presto in grado di soddisfare i requisiti energetici delle aziende. Secondo Forrester Research [10], nei prossimi anni il 60% dei data center si scontreranno con problemi di spazio, esigenze raffreddamento e limiti di alimentazione.

1.1.3 Green IT e ambiti

Green IT o Green Computing è un'espressione che include molteplici aspetti ma in generale può essere definito come *lo studio e la sua conseguente applicazione alla progettazione, produzione, uso e smaltimento di computer, server e sottosistemi associati, in modo efficiente e con il minor impatto ambientale possibile*. Il Green IT cerca inoltre di far coincidere interessi economici e miglioramenti delle performance con le responsabilità etiche e sociali che derivano dalle attuali condizioni ambientali precarie [11]. La complessità degli studi relativi all'impatto ambientale richiede un approccio olistico, ovvero un approccio scientifico che enfatizza lo studio dei sistemi complessi [12]. Da questo punto di vista un'altra interessante definizione è quella riportata in [2], che definisce il Green IT un *paradigma ingegneristico focalizzato su un sforzo globale e multiforme per la riduzione dei consumi energetici e la promozione della sostenibilità ambientale*.

Più in particolare, il termine Green IT può essere riferito a tre aree tematiche specifiche:

- l'efficienza energetica dell'IT
- la gestione eco-compatibile del ciclo di vita dell'IT
- l'utilizzo dell'IT come strumento per una green governance

La prima area si focalizza sull'efficienza energetica dell'IT, ossia sul rapporto tra prestazioni e consumo di energia. L'efficienza energetica può essere migliorata agendo sia sulla progettazione che sulla gestione delle infrastrutture e dei data center, nonché modificando la cultura aziendale e le pratiche di utilizzo. Per esempio, se si considera il TPC-C che, come verrà spiegato in seguito, è un benchmark comunemente utilizzato per valutare le prestazioni dei processori, l'efficienza energetica può essere misurata in migliaia di transazioni al minuto per Watt assorbito (Ktpm-c su Watt). Il valore di tale indice nell'ultimo decennio è aumentato di un fattore 2,5, indicando quindi un miglioramento dell'efficienza energetica. Tuttavia la crescente domanda di capacità di calcolo e l'aumento del consumo energetico dell'IT in termini assoluti impongono di migliorarne ulteriormente e in modo più radicale l'efficienza energetica, ovvero di rendere green l'IT [23]. Numerosi sono i fattori di inefficienza energetica in un sistema IT, così come sono numerose le leve su cui è possibile agire per migliorare.

Un approccio diffuso è il cosiddetto approccio multilivello [36], che consiste nell'agire su ciascuno dei diversi livelli infrastrutturali di cui è costituito un sistema informatico. In figura 1.11 è rappresentata la ripartizione dei consumi energetici dei componenti di un data center. Circa il 40% della potenza in ingresso viene assorbito dai sistemi HVAC ¹⁴ e apparati UPS ¹⁵,

¹⁴Heating, Ventilation and Air Conditioning

¹⁵Uninterruptible Power Supply

un altro 42% è assorbito da ventole, trasformatori AD/DC, storage e solo il 18% raggiunge i processori. Il fatto che i processori siano in idle durante alcuni periodi di funzionamento, implica un consumo di energia effettivo per le computazione di appena il 3% [9, 13].

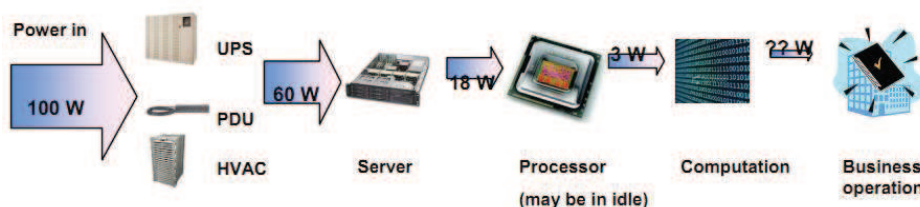


Figura 1.11: Ripartizione del consumo energetico di un data center

Un'analisi complementare a quella appena fatta, può essere svolta analizzando la figura 1.12.

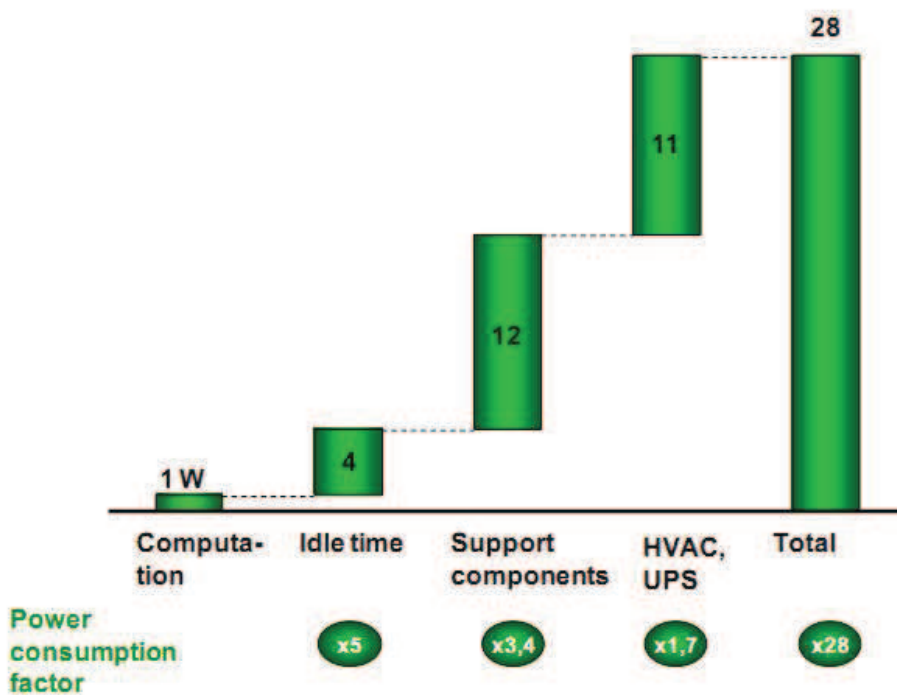


Figura 1.12: Energia assorbita da un data center (Watt, Source: Renzi, 2007)

Supponendo che un processore assorba 1W per elaborare l'informazione, l'energia totale consumata dal sistema può essere fino a 28 volte maggiore. Questo effetto amplificante è imputabile alla presenza dei drivers, alla ge-

stione della memoria, ai sistemi di raffreddamento e tutti gli altri sistemi di supporto necessari al processore. Investire dunque nell'ottimizzazione dei livelli più bassi del sistema, ad esempio a livello software, comporta dei benefici che si amplificano ai livelli superiori e hanno un grande impatto sul consumo energetico complessivo. Un software inefficiente richiederà più operazioni del processore che a sua volta richiederà più memoria, dovrà essere raffreddato di più e così via.

L'efficienza attuale di un sistema IT nel suo complesso è stata stimata come non superiore al 50% dell'efficienza massima teorica. La tabella 1.13 presenta alcune stime di massima sull'efficienza energetica dei vari livelli dei sistemi IT più comunemente utilizzati.

Livello	Efficienza energetica attuale stimata
Infrastruttura	50%
Sistema	40%
Server	60%
Microprocessore	0,001%
Software	20%
Rete	10%
Database	60%
Pratiche di utilizzo dell'IT	30%

Figura 1.13: Efficienza energetica stimata rispetto all'efficienza massima teorica degli attuali sistemi IT suddivisi per diversi livelli logici/strutturali

A livello di infrastruttura, le due maggiori cause di inefficienza sono i gruppi di continuità e i sistemi di condizionamento, che spesso sono a potenza fissa, mentre potrebbero essere regolati in modo dinamico per raffreddare dove necessario e quando necessario, a seconda del carico di lavoro effettivo del data center. La disposizione degli scaffali e delle bocchette di condizionamento è inoltre molto spesso stabilita in base a regole empiriche o dettate dall'esperienza, soprattutto nei data center di piccole dimensioni, quando invece studi termodinamici degli ambienti porterebbero a notevoli guadagni di efficienza. A livello di sistema, notevoli risparmi possono invece essere raggiunti bilanciando opportunamente i carichi di lavoro sui vari server. A livello di singolo server, molta energia è dissipata dalle periferiche e dai componenti ausiliari. In questo caso, piccoli accorgimenti possono portare a grandi risparmi di energia. Per esempio utilizzare ventole a velocità variabile può ridurre il consumo di energia delle ventole stesse fino al 45% e utilizzare meno drive più potenti può ridurre del 50% il relativo consu-

mo. Altri risparmi possono essere ottenuti dividendo la cache in segmenti alimentati solo quando necessario, o in alcuni casi particolari sostituendo parte degli hard-disk con memorie allo stato solido. Un'altra grande fonte di dispersione è la conversione dell'energia da alternata a continua a livello di singolo server: se la conversione fosse fatta da un solo alimentatore per tutto il data center, come avviene nei data center di alcune aziende di telecomunicazioni dove le macchine sono alimentate a 48V in continua, l'efficienza energetica sarebbe molto più elevata e le dispersioni di calore più contenute. Le inefficienze più grandi si hanno a livello dei processori, che si basano su chip ancora ben lontani dai computer quantistici che si avvicinano ai consumi minimi teorici previsti dalle leggi della fisica quantistica [23]. Molto spesso i processori non funzionano al massimo delle loro potenzialità e non sono sfruttati in modo efficiente. È stato dimostrato che abbassare la frequenza di clock e passare da processori single-core a quad-core può ridurre i consumi relativi anche del 50%.

Le operazioni eseguite dal processore sono in un certo senso guidate dal software, che a sua volta comprende vari livelli, dal sistema operativo al middleware e alle applicazioni utente. Molto spesso, la progettazione del software non tiene adeguatamente conto dell'efficienza energetica e il consumo di energia non viene neppure contemplato nel trade-off fra costi, prestazioni e qualità. Un recente filone di ricerca si occupa di investigare quali particolari aspetti della qualità interna del software abbiano un impatto sulla sua efficienza energetica.

A livello di rete, una maggiore efficienza energetica potrebbe essere ottenuta sia ripensando in ottica green gli algoritmi di routing, sia ottimizzando i dispositivi hardware per la gestione della rete stessa. Anche la qualità dei dati memorizzati nei database ha un impatto sull'efficienza energetica: la scarsa qualità dei dati (per esempio valori inesatti o non aggiornati) porta ad eseguire più transazioni ed operazioni del necessario con un conseguente dispendio di energia.

Infine, il modo in cui l'IT viene utilizzato (cioè le pratiche di utilizzo) può essere causa di ingenti sprechi di energia. Recenti ricerche hanno dimostrato che un corretto uso delle funzioni di power management (ormai presenti su tutti i moderni PC), un uso intelligente degli screen-saver e l'abitudine a spegnere i sistemi quando non in uso, possono portare a risparmi dell'ordine del 60%. In generale, per ottimizzare le performance energetiche di un sistema IT è necessario un cambiamento della cultura aziendale sia degli utenti che degli amministratori del sistema facendo loro comprendere che quasi sempre i risparmi energetici portano anche a risparmi di costo.

La seconda area tematica di cui si occupa il Green IT riguarda invece l'impatto ambientale del ciclo produttivo dei componenti IT e del loro smaltimento. L'inquinamento di cui l'industria IT è responsabile non è infatti dovuto solamente al consumo di energia, ma anche alle sostanze tossiche disperse nell'ambiente. Basti pensare che secondo alcune recenti ricerche [27]

il 70% dell'inquinamento del suolo da piombo, cadmio e mercurio deriva direttamente o indirettamente dall'IT. I rifiuti di apparecchiature elettriche ed elettroniche (RAEE), chiamati anche WEEE ¹⁶ o più semplicemente e-waste sono tutti i rifiuti derivanti da qualunque apparecchiatura elettrica od elettronica che viene dismessa in quanto guasta oppure obsoleta. Tali rifiuti, che contengono sostanze tossiche e non sono biodegradabili, costituiscono un rischio sempre maggiore per l'ambiente. La recente direttiva WEEE dell'Unione Europea (2002/95/CE), recepita in Italia dal Decreto RAEE (D.Lgs 151/2005) stabilisce precise norme per la raccolta differenziata e il recupero di tali rifiuti. La ricerca Green IT in questo campo si occupa di ridurre le sostanze inquinanti presenti nei componenti IT a partire dal processo produttivo, ottimizzare il packaging, eco-etichettare le varie unità, e approfondisce le diverse tecnologie di ricondizionamento e di recupero dei componenti dismessi. Come è possibile intuire, questa area è molto vasta e si intreccia con numerose altre discipline.

Infine, la terza accezione del Green IT va al di là dell'analisi dell'impatto ambientale dei sistemi IT ma si focalizza sull'utilizzo dell'IT come strumento per misurare e monitorare i parametri green (per esempio consumo energetico, temperatura, consumo di carta e materiali di consumo, rifiuti tossici prodotti) di tutti i processi di business, IT e non IT. Secondo una recente ricerca, l'86% dei dipartimenti ICT nel Regno Unito non conosce il peso delle proprie emissioni di CO₂, e l'80% delle organizzazioni non conosce la bolletta elettrica delle proprie attività. Una ricerca analoga, svolta dal Dipartimento di Elettronica e Informazione del Politecnico di Milano su 140 aziende italiane di piccole e medie dimensioni, ha rivelato che l'89% dei responsabili IT non conosce il consumo della propria strumentazione. È evidente come sia difficile ottimizzare ciò che non si conosce, così come è poco probabile che un responsabile IT sia incentivato ad investire risorse per diminuire i consumi energetici di apparati i cui costi non afferiscano al suo budget. Un corretto approccio al Green IT non può prescindere da una visione strategica all'interno dell'azienda, con i cambiamenti culturali e organizzativi che ne conseguono. La definizione di KPI ¹⁷ green da portare all'attenzione dei decisori d'impresa (consumo di energia, temperatura, CO₂, sostanze inquinanti ...) segmentandoli per le opportune dimensioni (su base temporale, per fase di lavorazione, per prodotto ...), può rivelarsi un passo decisivo per il monitoraggio e il miglioramento dell'efficienza energetica di un processo industriale. Così come avviene per altri parametri di efficienza ed efficacia, questi KPI possono essere memorizzati e analizzati tramite cruscotti direzionali e di business intelligence, opportunamente alimentati da sensori per la raccolta dei dati e reti intelligenti per l'aggregazione degli stessi tramite strumenti di data mining.

¹⁶Waste of Electric and Electronic Equipment

¹⁷Key Performance Indicator

1.2 Il Green Software

Ormai da qualche anno si parla diffusamente dell'efficienza energetica dell'IT, ma raramente si affronta il problema del ruolo rivestito dal software nel determinare il consumo energetico dell'IT. È infatti il software a “guidare” il funzionamento dell'hardware e quindi ad essere il primo responsabile del consumo [8]. Nei paragrafi precedenti si è discusso ampiamente sull'efficienza energetica nei data center, sottolineando come rappresenti un tema di ricerca ed innovazione su cui diversi centri di ricerca e aziende stanno lavorando ormai da diversi anni. L'aumento dell'efficienza energetica dell'hardware (in particolare server e processori) è il primo passo verso l'ottimizzazione dell'intero sistema IT e le grandi case produttrici di hardware svolgono da anni ricerche in questo senso. Si pensi ad esempio alla maggiore autonomia delle batterie dei dispositivi mobili o ai sistemi mainframe, il cui rapporto MIPS¹⁸ per Watt è cresciuto di un fattore 28 negli ultimi 30 anni. Tuttavia, benchè i dispositivi hardware siano fisicamente responsabili del consumo di energia, che viene dissipata sotto forma di calore, il responsabile primo di tale consumo è il software, che “guida” l'hardware e determina quali e quante operazioni elementari devono essere eseguite.

1.2.1 L'efficienza energetica nel Software

L'efficienza energetica ha sempre rappresentato un obiettivo nella progettazione dell'hardware ma raramente o quasi mai nel ciclo di vita del software o nei relativi strumenti e metodologie di sviluppo. La disponibilità di hardware sempre più efficiente e a basso costo ha causato, fino a pochi anni fa, un completo disinteresse da parte degli sviluppatori e della ricerca nei confronti dell'efficienza energetica del software.

Il green software è la disciplina che studia le modalità secondo cui il software influisce sui consumi energetici dell'IT e come ottimizzarle. Per comprendere meglio l'ambito di applicazione del green software può essere utile fare un paragone con il mondo automobilistico. Se si vuole consumare poca benzina per andare in auto da Milano a Torino la prima cosa da fare è quella di scegliere una macchina che percorra tanti km con un litro di benzina. Nel mondo informatico questo equivale ad utilizzare hardware efficiente, cioè in grado di effettuare tante transazioni elementari o flops¹⁹ con 1Wh di energia. Ci sono però tante altre soluzioni che si possono adottare per consumare meno benzina. Innanzitutto si può cercare di viaggiare a pieno carico. Per esempio se si è in 9 persone si può utilizzare un pulmino anziché 2 automobili. Questo equivale a bilanciare i carichi di lavoro e a virtualizzare i sistemi, pratiche che rientrano nell'ambito del green data center. Dopodiché si può viaggiare alla velocità che minimizza il consumo di carburante, che

¹⁸Million Instructions Per Second

¹⁹floating point operations

solitamente non coincide con la massima velocità raggiungibile dal veicolo. Infine, si può scegliere la strada che minimizza il numero di km percorsi. Queste due ultime tipologie di azioni corrispondono nel mondo informatico al green software. Un'applicazione può essere valutata in base alla sua efficienza energetica, oltre che in base agli altri classici parametri di merito, come il tempo di risposta. Inoltre, un'applicazione scritta "bene" permetterà di soddisfare i requisiti funzionali tramite il minor numero possibile di operazioni elementari e quindi di energia.

Mentre diverse ricerche sono state svolte sull'hardware, sui sistemi embedded [4] e sui data center, il tema dell'efficienza energetica del software applicativo è relativamente nuovo e inesplorato. Il ciclo di sviluppo del software non prende praticamente mai in considerazione l'efficienza energetica come obiettivo e non esistono, o hanno carattere per lo più sperimentale, modelli di sviluppo, best practices o strumenti che diano una misura oggettiva del consumo energetico del software. La letteratura attuale sull'ingegneria del software non fornisce neppure metriche per misurare l'efficienza energetica del software. In figura 1.14 è riportato il modello ISO/IEC 9126:2003, confluito poi nel sistema di norma ISO/IEC TR25000:2005.

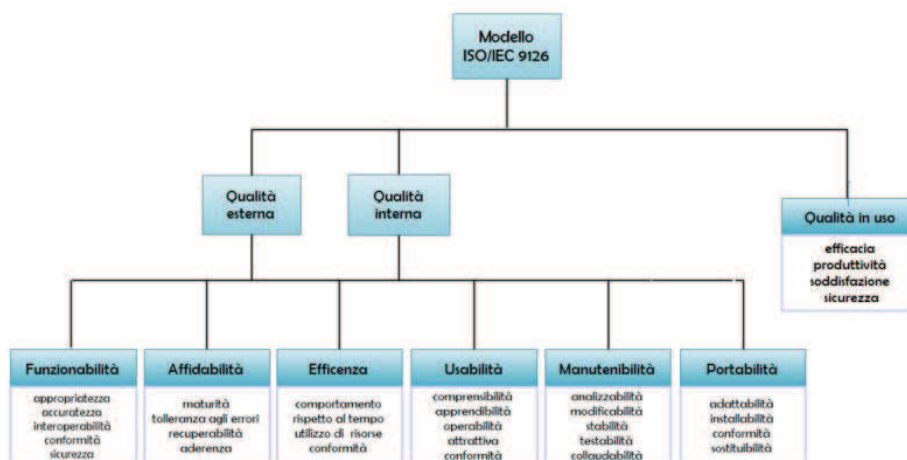


Figura 1.14: Modello ISO/IEC 9126

In questo modello sono definite 50 metriche di qualità del software ma, tra queste, non compaiono metriche relative all'efficienza energetica. In parte ciò è giustificato dalla crescente disponibilità di hardware efficiente a basso costo, che induce i programmatori a non occuparsi dell'efficienza energetica del codice.

Per quanto riguarda i sistemi embedded, precedenti studi si sono focalizzati sul Low Power Software, che si occupa di progettazione orientata al risparmio energetico per sistemi che hanno stringenti requisiti di consumo,

come ad esempio i portable device. La diminuzione del consumo, in questi dispositivi, permette di ridurre i limiti imposti dalla miniaturizzazione, dilatare i tempi tra le ricariche dei dispositivi, ed essere quindi competitivi sul mercato. I sistemi embedded sono tipicamente caratterizzati anche da ristretti limiti di spazio di memoria a disposizione. Una metodologia utilizzata per ovviare a questo problema è la *code compression*, cioè la compressione del codice sorgente che riduce la quantità di memoria richiesta e, necessitando di un numero minore di accessi alla main memory, riduce la dissipazione di energia nel bus e nelle interconnessioni. Su questo tipo di dispositivi, diventa fondamentale anche il *task scheduling*, considerato uno dei metodi più comuni per ottenere una riduzione del consumo energetico. In particolare, nei light-weight embedded systems, lo scheduling risparmia energia spegnendo automaticamente i dispositivi non operativi in un dato momento. Anche l'ordinamento dell'esecuzione dei tasks permette una miglior gestione dei tempi d'idle ed un minore spreco d'energia. Nei dispositivi mobili, l'unità hardware che consuma di più è la CPU. A tal proposito, sono state sviluppate diverse tecniche che permettono una riduzione dello spreco energetico implementando il *DVFS* ²⁰. Questa tecnica permette di modificare dinamicamente il voltaggio della CPU per garantire la minima energia necessaria affinché la frequenza del circuito possa processare il workload di sistema, rispettando i vincoli di tempo/throughput imposti e riducendo il consumo energetico dipendente in modo quadratico dal livello di voltaggio fornito al sistema.

Per quanto riguarda applicazioni end-user invece, non molto è stato fatto nel campo del green software design nonostante sia stato mostrato come l'ottimizzazione energetica del software abbia ancora ampio spazio di miglioramento e, soprattutto, come notevoli benefici tangibili in termini di diminuzione dei costi possano essere apportati. Una differenza sostanziale tra software embedded e MIS ²¹ è l'obiettivo per il quale sono progettati: mentre i requisiti principali di un software embedded sono relativi alle performance, applicativi end-user quali software bancari o ERP ²² devono innanzitutto rispondere ai requisiti funzionali. Lo sviluppo di software MIS fa uso di linguaggi di alto livello, librerie e framework di sviluppo che, da un lato riducono la complessità di sviluppo, ma dall'altro impattano sul comportamento dell'applicazione stessa.

La maggior parte dei manager dell'IT sembra credere che il software di per sé abbia un impatto molto limitato sui consumi, specialmente nell'ambito dei MIS. È generalmente accettato che i sistemi operativi influenzino le prestazioni e i consumi di energia, ma non le applicazioni. Queste convinzioni raramente sono supportate da un approccio scientifico e da misure

²⁰Dynamic Voltage and Frequency Scaling

²¹Management Information Systems

²²Enterprise Resource Planning

sperimentali. In realtà il potenziale risparmio ottenibile dal green software è elevato. Nella maggior parte dei data center infatti, la percentuale di energia entrante, effettivamente utilizzata per il calcolo, è molto bassa a causa delle inefficienze dei diversi livelli infrastrutturali. Questi dati possono essere interpretati al contrario: per ogni Watt di energia utilizzato per il calcolo sono necessari almeno 5 o più Watt totali. Quindi i risparmi ottenuti a livello software, mediante l'impiego di tecniche di programmazione energeticamente più efficienti che riducono i carichi computazionali destinati al sistema, vengono amplificati da tutti i livelli infrastrutturali soprastanti: se vengono eseguite meno operazioni elementari sarà prodotto meno calore, quindi sarà necessaria meno energia per il condizionamento, e così via.

Lo studio condotto da Formenti e Gallazzi presso il Dipartimento di Elettronica ed Informazione del Politecnico di Milano, confermano il significativo impatto del software sul consumo energetico [29].

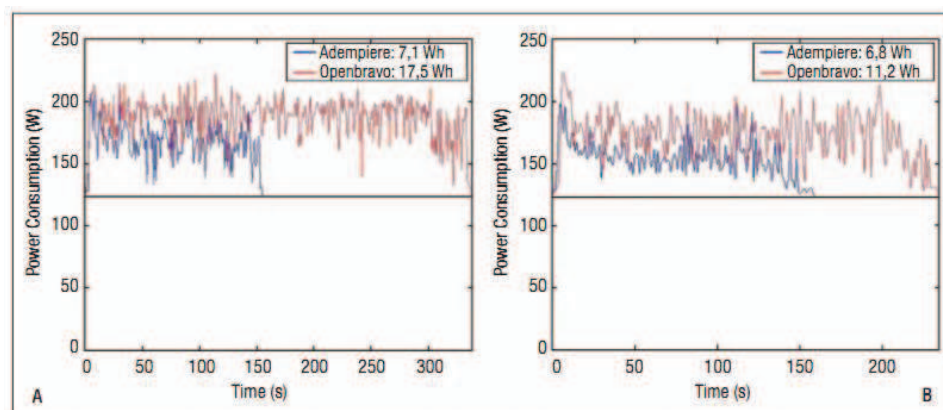


Figura 1.15: Potenza consumata nei sistemi ERP

I grafici 1.15, 1.16 e 1.17 mostrano la potenza consumata nel tempo per tre diverse categorie di software (ERP, CRM e DBMS) eseguite sulla stessa piattaforma e configurate in maniera tale da risultare il più possibile confrontabili. La linea dritta in ciascun grafico rappresenta il valore di potenza assorbito in idle dal sistema mentre, nel riquadro in alto a destra, vengono indicati i valori dell'energia totale (aggiuntiva rispetto all'idle) assorbita da ciascuna applicazione per completare l'esecuzione del carico di lavoro. L'energia consumata è calcolata integrando i valori della potenza assorbita nel tempo impiegato. Il primo importante risultato che emerge da questi grafici è che la potenza consumata dal server quando esegue l'applicazione è significativamente più alta rispetto a quella assorbita in idle (fino al 72% in più). Emerge inoltre un altro fatto importante: applicazioni diverse appartenenti alla stessa categoria richiedono al sistema di consumare quantità di energia molto diverse per soddisfare gli stessi requisiti funzionali. Un'osservazione critica dei dati proposti suggerisce dunque come l'efficienza energetica

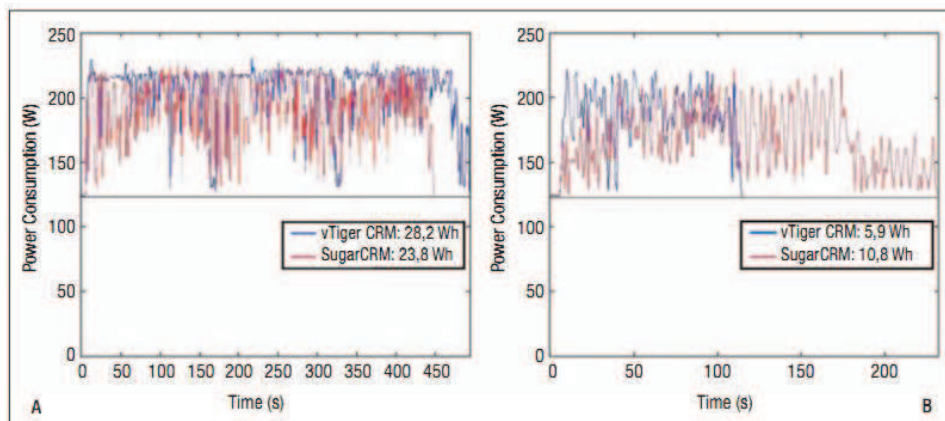


Figura 1.16: Potenza consumata nei sistemi CRM

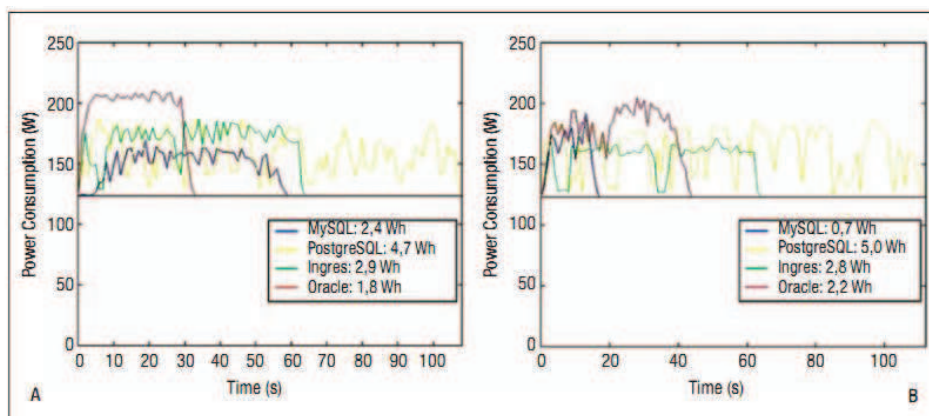


Figura 1.17: Potenza consumata nei DBMS

debba essere considerata come un nuovo parametro di merito di un'applicazione software, in aggiunta alle performance. La performance temporale infatti non è sufficiente a giustificare la diversità di efficienza energetica delle applicazioni software.

Attualmente, gli studi condotti al Politecnico di Milano sul green software prevedono due filoni di ricerca:

- metriche per la misurazione dell'efficienza energetica del software
- ottimizzazione dell'efficienza energetica del software

Per quanto riguarda il primo filone è importante distinguere il concetto di efficienza da quello di consumo: l'efficienza è data dal rapporto tra il consumo energetico e la quantità di lavoro svolto. Un set di metriche per l'efficienza energetica del software può essere utile sia nella fase di software selection, integrando i parametri di merito attualmente utilizzati, sia nella fase di software evaluation, con conseguente valutazione anche del team di sviluppo. Esse costituiscono inoltre lo strumento essenziale per fare benchmarking, ovvero poter confrontare tra loro sistemi diversi. Esistono già diverse metriche consolidate per misurare l'efficienza energetica dell'IT a vari livelli.

Abbiamo già parlato del PUE (Power Usage Effectiveness) utilizzato per valutare l'efficienza dei livelli infrastrutturali. Per valutare il bilanciamento dei carichi e il livello di virtualizzazione si possono adottare metriche come la percentuale media di utilizzo dei processori, sia fisici che virtuali. Vi sono poi diverse metriche utilizzabili a livello di server e di processore per valutare l'efficienza energetica dell'hardware, per esempio W/tpm (Watt per transazioni al minuto), FLOP/Wh (operazioni floating point per Wh) oppure MIPS/W (Milioni di istruzioni al secondo per Watt) per gli ambienti mainframe.

Attualmente non vi sono ancora metriche consolidate e sufficientemente generali per misurare l'efficienza energetica del software. A tal proposito Galli, in [31], individua cinque categorie di metriche usate come proxy per il consumo energetico:

metriche innovative sul linguaggio macchina lavorano sul codice macchina e misurano direttamente ciò che viene eseguito.

metriche classiche di qualità del design appartenenti alla Suite CK che valuta la complessità di un sistema software progettato secondo il paradigma object-oriented.

entropia dell'espressività sul codice sorgente valuta quanto un determinato frammento di codice sfrutti le potenzialità espressiva del linguaggio di programmazione che utilizza.

metrica Green sul codice macchina valuta alcune informazioni semantiche attraverso la definizione di un peso per ogni microistruzione ad un certo livello di annidamento del codice.

metrica Green sul bytecode come la precedente sebbene si focalizzi sul linguaggio Java

Nello studio condotto da Vagnato e Amalfi [35], si è invece analizzato l'impatto energetico derivante dall'uso o meno di librerie esterne, durante lo sviluppo di un software. I risultati hanno portato alla definizione di tre metriche ad-hoc che in un certo senso sono un indice di come è stato progettato e implementato il software.

EFL questa metrica si basa sul conteggio delle linee di codice dei metodi esterni richiamati dalla classe "main" dell'applicazione normalizzato sul totale delle linee di codice del software.

EFC la seconda metrica si basa semplicemente sul conteggio dei metodi esterni richiamati dalla classe "main" dell'applicazione normalizzato sul totale dei metodi del software.

EFLC la terza e ultima metrica è una conseguenza delle metriche EFL e EFC

Per quanto concerne il secondo filone di ricerca, ovvero l'ottimizzazione dell'efficienza energetica del software, gli approcci perseguibili possono essere raggruppati in due grandi categorie:

- ottimizzazione a livello di codice
- ottimizzazione a livello di sistema

Un primo approccio è quello di ottimizzare il codice: scrivere meglio il codice può avere un effetto sulle performance energetiche dell'applicazione. Ovviamente, queste tecniche sono applicabili solamente al software sviluppato inhouse e possono tradursi sia in metodologie, linee guida e tool per lo sviluppo, che in linee guida a livello organizzativo e relative alle competenze e al percorso di formazione degli sviluppatori. In generale, l'utilizzo di livelli di astrazione e di ambienti di sviluppo user-friendly è molto diffuso nei moderni approcci allo sviluppo del software, in quanto semplifica la programmazione e quindi riduce il costo di sviluppo. I livelli di astrazione, le librerie, i componenti standard e i framework di sviluppo permettono di impiegare, per lo sviluppo delle applicazioni, programmatori non specializzati che possono essere formati per l'uso di quei particolari strumenti e componenti preconfezionati in poco tempo e con un investimento ridotto. Queste applicazioni soddisfano i requisiti funzionali, ma la generalità che deve caratterizzare gli strumenti utilizzati non può che inficiarne le prestazioni, come suggerito dai risultati empirici in [29]. Se le stesse applicazioni

fossero sviluppate da programmatori esperti, capaci di scrivere codice ottimizzato specifico per il contesto di utilizzo, l'efficienza risultante sarebbe probabilmente maggiore. Ovviamente l'impiego di bravi programmatori fa aumentare considerevolmente i costi di sviluppo e potrebbe alzare anche i costi di manutenzione, in quanto il codice potrebbe essere più difficile da interpretare e modificare. Occorre quindi valutare il TCO esteso di un'applicazione e bilanciare i vari fattori per determinare la strategia di sviluppo ottima. Ad oggi la ricerca non ha ancora prodotto indicazioni chiare in questo senso, ma è molto probabile che un approccio 80-20 si possa rivelare vincente: ottimizzare tramite programmatori specializzati il 20% del codice responsabile dell'80% dei consumi, puntando invece sulla facilità di sviluppo della restante parte.

Migliorare l'efficienza energetica ottimizzando il codice potrebbe però non essere sempre fattibile oppure non essere conveniente. Se da una parte potrebbe essere sensato riscrivere e ottimizzare le routine esigenti dal punto di vista energetico o più comunemente eseguite in un sistema, dall'altra non sarebbe pensabile realizzare l'analisi e l'ottimizzazione di tutto il codice del sistema informativo di una banca e, in ogni caso, lo sforzo necessario per questa operazione difficilmente sarebbe ripagato dai risparmi ottenuti. Per questo motivo è opportuno sviluppare anche strumenti di ottimizzazione a livello di sistema, applicabili cioè in modo trasversale e senza che sia richiesto operare a livello di codice. Molti sono gli strumenti e le azioni che possono essere sviluppati a questo livello e la ricerca è quanto mai aperta. Un esempio di azione, riprendendo le evidenze empiriche riscontrate da Formenti e Gallazzi in [29], potrebbe essere una scelta oculata dello stack applicativo, considerando l'impatto che l'interazione fra sistema operativo e applicazione ha sul consumo complessivo. In generale la riduzione del peso dello stack applicativo libera risorse, ottenendo un controllo a granularità più fine di quelle effettivamente utilizzate. Questo tipo di azione è utilizzata principalmente nella scrittura di nuovo codice e in particolare per applicazioni critiche, in modo che il costo aggiuntivo di sviluppo e manutenzione sia più che bilanciato dai benefici ottenuti in termini di risparmio energetico, garantendo quindi una complessiva diminuzione del TCO. Al contrario, per grandi basi di codice preesistenti, in cui il costo di manutenzione può risultare il fattore critico, è necessario trovare leve diverse. Sono desiderabili in questo caso metodologie di ottimizzazione almeno semiautomatica, applicabili a livello di sistema sull'intera base di codice da parte di personale che abbia competenze di tipo sistemistico, e che non richiedano quindi modifiche manuali al codice, ma piuttosto la regolazione di parametri di ottimizzazione. A tal proposito due tipologie di intervento a diversi livelli sono la regolazione della gestione della memoria e la memoizzazione semiautomatica al livello dell'applicazione [8].

1.3 Benchmark per l'efficienza energetica

Negli ultimi anni, la necessità delle aziende di una maggiore trasparenza dei costi operativi associati ai consumi energetici, ha spinto i principali consorzi di benchmarking a sviluppare nuove metodologie per la misurazione dell'efficienza energetica dei data center. Nonostante la ricerca si stia tutt'ora impegnando nella definizione di una metrica universale, le organizzazioni fornitrici di benchmark hanno fornito in questi anni alcune estensioni per la valutazione olistica dell'efficienza energetica dei sistemi IT [20].

Quella che segue è un'analisi delle due principali organizzazioni per il benchmarking, impegnate nella definizione di metriche e metodologie per la misurazione dell'efficienza energetica. In particolare, i consorzi analizzati, sono:

- The Transaction Processing Performance Evaluation Council (TPC)
- The Standard Performance Evaluation Corporation (SPEC)

SPC (Storage Performance Council) è un'ulteriore corporation non-profit che ha recentemente definito SPC-1/E, una specifica di benchmark per l'efficienza energetica di sottosistemi di storage, DBadapter, controllers e reti SAN²³. Si rimanda a [18] per ulteriori approfondimenti.

1.3.1 Il Benchmark TPC-Energy

Il Transaction Processing Performance Council è un consorzio non-profit di aziende, fondato nel 1988 con lo scopo di definire delle suites di benchmark per sistemi transazionali. Una tipica transazione misurata e testata da benchmark TPC appartiene ad un processo di business ed include una o più interazioni con i sottosistemi di storage o basi di dati. Esempi di transazioni includono il controllo di inventario, la prenotazione di un volo, una transazione bancaria . . . I benchmark TPC si rivolgono a sistemi OLTP²⁴ complessi, composti da numerosi sottosistemi quali lo storage, i server, sistemi di networking.

Il TPC attualmente propone la specifica per i seguenti benchmark:

- TPC-C: benchmark per sistemi a supporto delle attività di gestione degli ordini
- TPC-E: benchmark per sistema OLTP complesso
- TPC-H: benchmark per sistemi DSS²⁵

²³Storage Area Network

²⁴On Line Transaction Processing

²⁵Decision Support System

Nel Giugno 2010 inoltre, il consiglio ha rilasciato la versione 1.2.0 della specifica TPC-Energy che estende tutti i benchmark TPC con la definizione di metriche di efficienza energetica. TPC-Energy consente ai produttori di fornire metriche nella forma “Watt su performance”, dove un'unità di performance dipende dal particolare benchmark TPC in uso. Attraverso la pubblicazione dei risultati di benchmark, le aziende sono in grado di valutare e configurare i sistemi che rispondono ai propri requisiti di costo, performance e di efficienza energetica.

Le metriche TPC-Energy sono direttamente correlate alle metriche TPC di performance. Per dare un'idea di tale correlazione è utile rappresentare le metriche TPC su tre assi, come mostrato in figura 1.18.

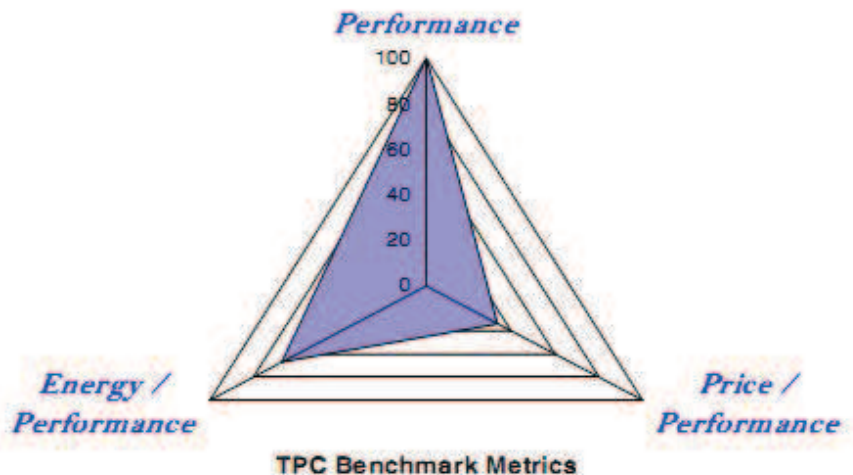


Figura 1.18: Metriche TPC

Il piano definito da tre punti sugli assi è un indicatore della “bontà” del sistema testato (SUT). Come è facilmente intuibile, il caso ideale è rappresentato dalla degenerazione del piano alla retta giacente sull'asse delle performance. Nei casi reali, una configurazione di sistema orientata all'efficienza energetica presuppone massimizzare la metrica di efficienza, accettando una parziale diminuzione delle performance e un aumento del costo per performance [19].

La specifica TPC-Energy descrive in dettaglio la metodologia di misurazione delle performance energetiche. Di seguito vengono brevemente descritti i principali componenti definiti dalla specifica che intervengono nel processo di misurazione:

Reported Energy Configuration (REC) : consiste in tutti i componenti che partecipano al processo di misurazione dell'energia. Può essere suddiviso in sottosistemi, ognuno dei quali comprende una o

più unità misurabile (Power Power Measurable Units - PMU). In figura 1.19 è riportato un esempio di REC, costituito da 3 sottosistemi (application server, database server, e storage)

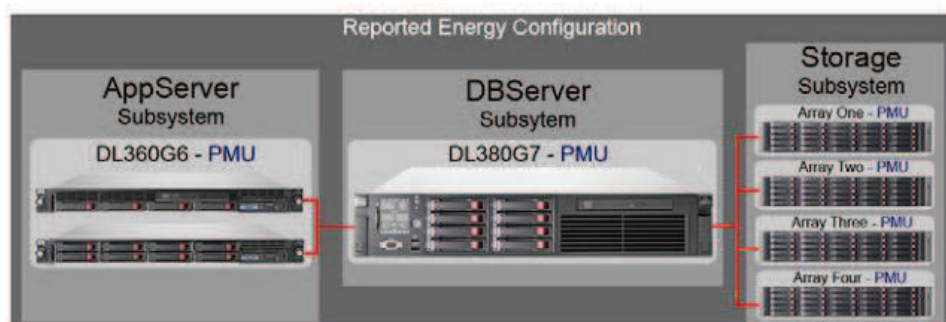


Figura 1.19: Reported Energy Configuration

Il sottosistema application server consiste in una singola PMU per la misura della potenza assorbita dai server su cui è installato il software di interfaccia tra gli utenti simulati e il database server. Alla stessa modo, il sottosistema database server è costituito da una singola PMU per la misura del consumo energetico del database server, su cui è installato il DBMS²⁶. Il sottosistema di storage infine è costituito da 4 PMU.

Power Measurable Unit (PMU) : la specifica TPC-Energy definisce una PMU (figura 1.20) come un componente o collezione di componenti del REC che possono essere singolarmente misurati dal Power Analyzer attraverso il software di benchmark (EMS).

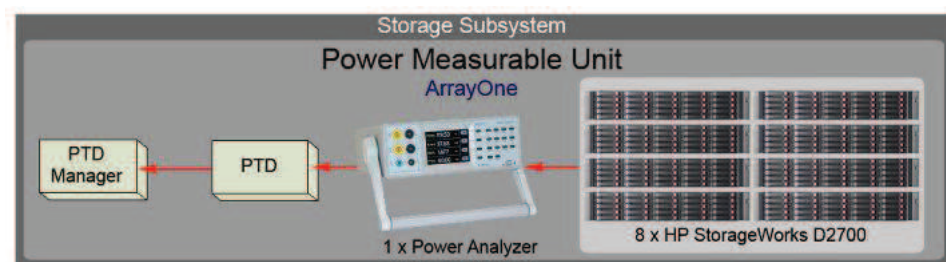


Figura 1.20: Power Measurable Unit

Power Analyzer Configuration : definisce i requisiti del Power Analyzer, il device connesso al SUT e responsabile dell'acquisizione dei dati energetici (potenza, voltaggio, corrente, fattore di potenza) raccolti durante l'esecuzione del benchmark. La configurazione include il settaggio del range e dell'accuratezza delle misure.

²⁶Database Management System

Energy Measuring System (EMS): package software che implementa il benchmark. Consiste a sua volta in tre moduli (figura 1.21):

- EMS Controller: modulo responsabile dell'orchestrazione dei vari componenti coinvolti nel processo di misurazione.
- PTD (Power Temperature Daemon): è una versione dello SPEC PTDaemon tool (descritto nel paragrafo seguente) customizzato per rispondere ai requisiti della specifica TPC-Energy. Comunica direttamente con il Power Analyzer e invia i dati misurati al PTD Manager.
- PTD Manager: modulo responsabile della sincronizzazione, logging e inoltro delle misure di potenza e temperatura ricevuti dal PTD al Controller.

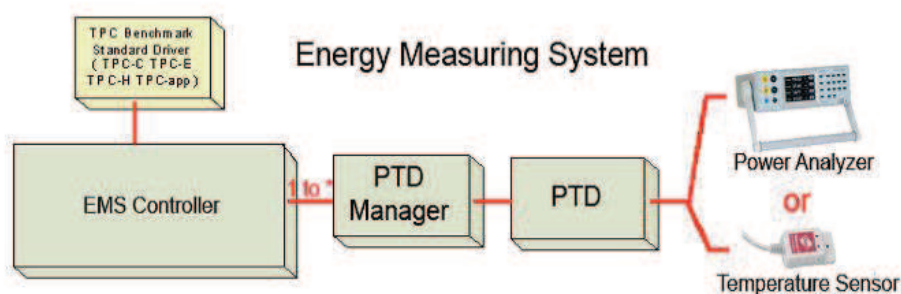


Figura 1.21: EMS

Report Generator : produce il report, in un formato standard, derivato dalla configurazione e dai dati di misura collezionati al termine dell'esecuzione del benchmark.

1.3.2 Il Benchmark SPECpower_ssj2008

SPEC (Standard Performance Evaluation Corporation) è un consorzio non-profit istituito nel 1988, il cui obiettivo è la definizione e il mantenimento di un insieme di benchmark applicabili alle più nuove generazioni di computer ad alte prestazioni. Attualmente il consorzio ha sviluppato più di 30 tipologie di benchmark industriali per la valutazione delle prestazioni in diverse aree applicative e fornisce licenze per il loro utilizzo non solo ad aziende, ma anche centri di ricerca e istituti di istruzione. Sono più di 10.000 i report approvati e pubblicati dal SPEC, compliant con le specifiche definite dallo stesso consorzio ("SPEC's General Availability and Fair Use rules").

SPECpower_ssj2008 è il benchmark SPEC, sviluppato a partire dal 2006, dedicato alla misurazione dei consumi energetici di sistemi server. Promosso da aziende membre quali AMD, Dell, Fujitsu Siemens Computers, HP,

IBM, Intel e Sun Microsystem, SPECpower_ssj2008 rappresenta una soluzione per le aziende che vogliono testare le performance energetiche dei propri sistemi in maniera relativamente facile, richiedendo un setup semplificato e riducendo così la possibilità di errore [22]. Il documento “SPEC Power and Performance Methodology” spiega dettagliatamente come configurare l'ambiente di test per diverse tipologie di sistemi server e descrive l'impatto di fattori ambientali quali ad esempio la temperatura, la pressione dell'aria e l'umidità.

L'architettura di una semplice implementazione del benchmark SPECpower_ssj2008 è riportata in figura 1.22. Come si osserva, il benchmark

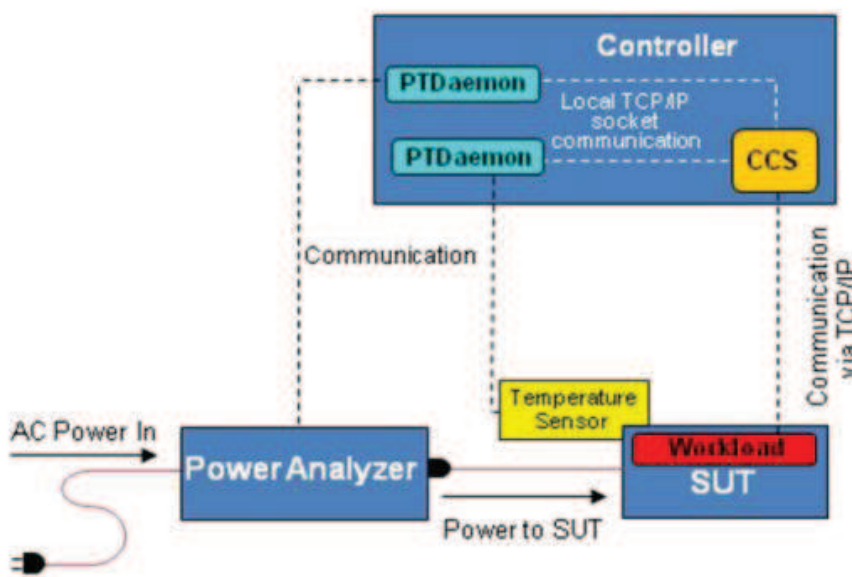


Figura 1.22: Architettura del benchmark SPECpower_ssj2008. Fonte: SPECpower_ssj2008 User Guide

consiste di tre componenti software principali:

- Server Side Java (SSJ) Workload: programma java che genera ed esegue un mix di transazioni con l'obiettivo di stressare il funzionamento della CPU, della memoria, la JVM, compilatori JIT ²⁷ e altri aspetti del sistema operativo del SUT.
- Power and Temperature Daemon (PTDaemon): si occupa del controllo dei sensori di potenza e temperatura per non alterare le misure del SUT.

²⁷Just in Time

- Control and Collect System (CCS): applicazione multi-thread java che gestisce i dati provenienti da sorgenti diverse (ad esempio workload in esecuzione su un altro SUT). Include il Visual Activity Monitor (VAM), package per la visualizzazione delle attività di più SUT simultaneamente.

Il benchmark, una volta fatto partire, avvia una fase di calibrazione che determina il throughput massimo del sistema, definito come il numero di transazioni completate al secondo su un intervallo di tempo fissato. In figura 1.23 è riportata un'esecuzione del benchmark SPECpower_ssj2008. Le performance e il consumo energetico sono calcolati per diversi carichi,

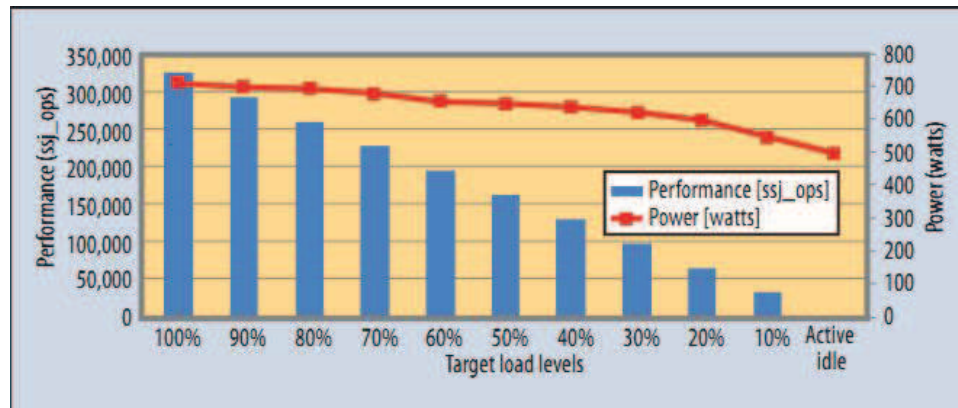


Figura 1.23: Consumo energetico a diversi carichi.

partendo da un valore pari al throughput massimo e decrementando ogni volta del 10% fino all'idle.

SPECpower_ssj2008 riporta l'efficienza energetica del sistema in termini ssj_ops eseguite per ogni watt consumato. Questa metrica è calcolata sommando le performance (espresse in ssj_ops) misurate ad ogni livello di carico, e dividendo tale somma per la potenza complessiva mediamente consumata (in Watt) incluso l'idle. Con riferimento all'esempio in figura 1.23 si riporta il calcolo della metrica SPECpower_ssj2008:

$$\frac{\sum Performance}{\sum Power} = \frac{1,799,250ssj_ops}{6,510watt} = 275 \frac{ssj_ops}{watt} \quad (1.1)$$

La prima release del benchmark, rilasciata nel 2007, includeva il supporto per singoli server. Nell'aprile del 2009 è stata rilasciata la versione SPECpower_ssj2008 1.10 per sistemi multi-node di server blade. Recentemente (settembre 2011) è stata rilasciata la versione 1.11.

1.3.3 Limite degli Energy-Benchmark

E' importante analizzare l'efficienza energetica di un sistema informativo a tutti i suoi livelli, in quanto gli attori coinvolti possono essere molto diversi. Se consideriamo un data center infatti, il PUE può essere influenzato da chi gestisce il data center, così come il livello medio di utilizzo dei processori. Le prestazioni a livello di hardware dipendono esclusivamente dai fornitori di tecnologia e quindi possono essere influenzate solo in fase di acquisto. Infine, l'efficienza energetica del software dipende dai clienti del data center, cioè da chi scrive o seleziona le applicazioni software da acquisire. La difficoltà nella definizione delle metriche di efficienza energetica del software risiede nella quantificazione del lavoro svolto da un'applicazione con una quantità unitaria di energia. Riprendendo il paragone relativo al mondo automobilistico, se l'efficienza di un'automobile può essere misurata in chilometri percorsi con un litro di benzina, nell'ambito del green software non è facile misurare i chilometri, mentre, è relativamente semplice misurare i litri di benzina consumata, ovvero l'energia. Questo sposta il problema alla definizione di concetti di transazione o di carichi di lavoro standard, il più possibile generali, da far eseguire ad un'applicazione per effettuare le misure.

Nei paragrafi precedenti abbiamo visto come i benchmark TPC, ad esempio, definiscano i carichi di lavoro standard, ovvero le transazioni, su cui poi è costruita la metrica di efficienza. Il problema principale dei benchmark descritti in precedenza risiede nel fatto che sono finalizzati al testing di un intero sistema piuttosto che l'applicazione finale. Per alcune applicazioni la definizione di transazione è abbastanza intuitiva (per esempio per i DBMS), mentre per altre, come gli ERP, occorre considerare differenti tipologie di transazione e quindi diverse metriche di efficienza.

Il lavoro di questa tesi mira a proseguire gli studi sull'identificazione e validazione di metriche proxy discusse in precedenza e la progettazione di un benchmark specifico per la misurazione dell'efficienza energetica del software, attraverso l'identificazione di workload tipici e la definizione di un insieme di transazioni standard per diverse categorie di software. Metriche basate sul codice, come quelle citate in precedenza, sono molto comode in quanto risultano misurabili tramite opportuni strumenti software semplicemente analizzando il codice dell'applicazione, senza bisogno di eseguirla. Le categorie di software a cui siamo interessati sono i sistemi gestionali (ERP, CRM) e l'obiettivo è giungere ad una metrica unica per ciascuna categoria. È evidente come la definizione e il calcolo di queste metriche sia più complicata che per gli altri livelli infrastrutturali. Ad esempio occorre considerare i dati di input e le dimensioni dei database sottostanti (dal punto di vista del consumo energetico è la stessa cosa modificare la giacenza a magazzino di un prodotto quando ci sono 10 tipologie di prodotti presenti o quando ce ne sono 1.000?).

Capitolo 2

Metodologia

Negli ultimi anni l'impatto dei consumi energetici sui costi operativi dei datacenter é diventato sempre piú rilevante. Investire in tecnologie green rappresenta oggi un'esigenza imprescindibile a cui le grandi aziende non possono piú rinunciare. Tuttavia, la ricerca nell'ambito dell'efficienza energetica dei sistemi IT si é sempre concentrata sull'hardware e sui sistemi embedded e poco o nulla sul software. L'opinione diffusa tra i manager dell'IT é che il software abbia un impatto limitato sui consumi energetici soprattutto per quanto riguarda i sistemi transazionali user-oriented come i sistemi enterprise. [33]

Se da un lato é accettata l'assunzione che i sistemi operativi influenzino i consumi energetici di un sistema informatico, dall'altro si trascura l'impatto dei software applicativi, nonostante siano questi progettati secondo metodologie e tecnologie molto diverse.

In questo capitolo si presenta lo studio condotto relativamente al consumo energetico di un'infrastruttura software enterprise. La presentazione parte da una descrizione ad alto livello dell'approccio di misura seguito, a cui segue la descrizione dettagliata del modello adottato per la stima del consumo energetico e la definizione di un benchmark per la misura e classificazione dell'efficienza energetica di tre sistemi enterprise open-source.

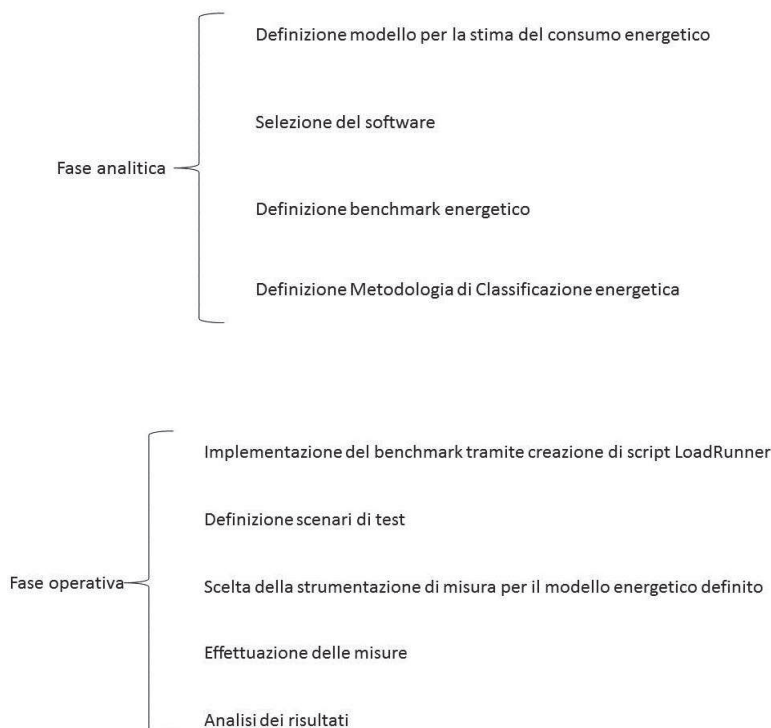
2.1 Piano di Misura

Il piano di misura si suddivide in due macrofasi:

- Macrofase analitica
- Macrofase operativa

Nella macrofase analitica si é definito il modello per la stima indiretta del consumo energetico. Segue poi la definizione di un benchmark per il consumo energetico di software enterprise open source, basato sulla specifica TPC-C.

La seconda macrofase invece, si pone come obiettivo l'implementazione di quanto definito nella macrofase analitica, al fine di svolgere il successivo lavoro di validazione del modello attraverso le diverse misure che si andranno ad effettuare. Entrambi le macrofasi si articolano lungo diverse attività che verranno spiegate in dettaglio nelle prossime sezioni e che riassumiamo qui brevemente:



2.2 Stima del consumo energetico

In questa sezione si descrive il modello che andremo ad usare per stimare il consumo energetico del software. Dopo la definizione del modello e della metodologia su cui esso si basa, segue la descrizione operativa di come si è utilizzato tale modello e attraverso quali strumenti si sono effettuate le misure, i cui risultati sono riportati nel capitolo successivo.

2.2.1 Modello energetico e stima indiretta del consumo

Il cuore di questo lavoro riguarda l'identificazione di un modello per il consumo energetico in grado di stimare il consumo di potenza in maniera indiretta, ovvero partendo dal consumo di risorse.

Ciò si rende necessario in quanto stiamo cercando di misurare e classificare

il software, e non l'hardware, dal punto di vista energetico, quindi abbiamo bisogno di trovare una correlazione tra quanto un'un'applicazione impegna risorse hardware e quanto effettivamente ciò si traduce in consumi energetici. Per capire meglio il contesto in cui ci mettiamo, è utile introdurre la problematica attraverso un semplice paragone automobilistico.

Immaginiamo di dover andare, con una automobile, da Roma a Milano. Esisteranno molti percorsi alternativi che, con distanze diverse, ci permetteranno di raggiungere la destinazione consumando un certo quantitativo di carburante. La lunghezza del percorso, ovvero i chilometri percorsi dall'automobile, rappresenta il consumo di risorse necessario. Ogni percorso ha però un costo che, nel nostro esempio, può essere quantificato dal consumo medio in litri di carburante per chilometro, moltiplicato per la distanza del percorso stesso. Oltre alla distanza infine, il consumo di carburante è dettato anche dalla velocità tenuta dall'automobile lungo il percorso. Ovviamente a parità di percorso, un'automobile consumerà tanto meno quanto più lentamente percorrerà il tragitto. Se riportiamo l'esempio nell'ambito del consumo energetico del software, è facile associare il server all'automobile e un programma che deve essere eseguito alla distanza da percorrere. Esistono infatti molteplici programmi che realizzano lo stesso task. Quello che cambia è il consumo di risorse (ad esempio l'utilizzo del processore) durante la sua esecuzione e il modo in cui queste vengono consumate (ad esempio la frequenza di clock del processore).

L'obiettivo che si vuole tentare di raggiungere è confermare sperimentalmente un'idea abbastanza intuitiva: un software la cui esecuzione comporta un basso consumo di risorse determina un consumo energetico inferiore rispetto ad un software con elevati requisiti in termini di risorse consumate. Così facendo si è dunque spostato il problema dalla misura diretta del consumo energetico, alla misura del consumo di risorse.

Il modello energetico proposto, considera come unica risorsa consumata dal software il processore. Al fine di quantificarne il consumo, si è fissata come metrica la percentuale di utilizzo che, a differenza di altre, è facilmente misurabile tramite strumenti software, come verrà spiegato nel seguito. La metrica, che in seguito si indicherà con γ , quantifica l'utilizzo del processore durante l'esecuzione di un'applicazione software ed è data dalla relazione:

$$\gamma = \int U_{\%CPU} dt$$

Una seconda metrica necessaria per rendere il modello completo, riguarda la misura di quanto è efficiente la risorsa dal punto di vista energetico. Tornando all'esempio dell'automobile, un'auto a bassi consumi è certamente più efficiente di un'auto ad elevati consumi. Nel caso di un server possiamo quantificare le performance energetiche del processore stimando la variazione di potenza assorbita a fronte di una variazione percentuale unitaria di utilizzo.

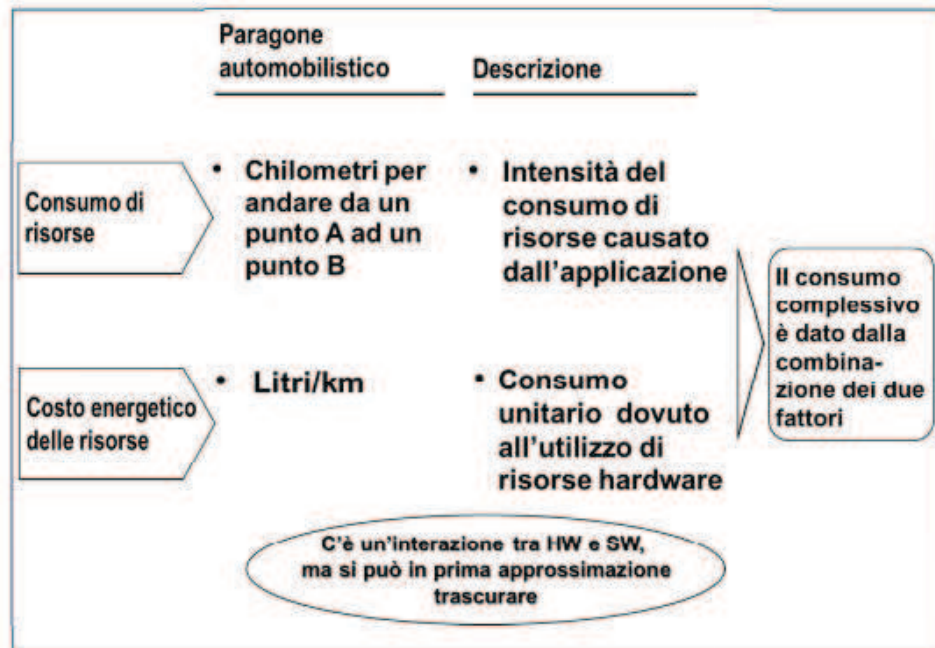


Figura 2.1: Paragone automobilistico

Questa seconda metrica verrà nel seguito indicata con β . Il modello risultante per la stima del consumo energetico di un sistema durante l'esecuzione di un'applicazione software é:

$$\begin{aligned}
 E_{sys} &= E_{idle} + E_{aggiuntiva} = \\
 &= E_{idle} + \Delta E_{cpu} + \Delta E_{others} = \quad (2.1) \\
 &= E_{idle} + \gamma\beta + \Delta E_{others}
 \end{aligned}$$

dove l'energia in idle é l'energia consumata dal sistema quando il processore non esegue l'applicazione.

Un paio di osservazioni sul modello appena presentato:

1. Dimensionalmente la relazione 2.1 é corretta. La metrica γ infatti ha dimensioni pari al prodotto $\%utilizzo * s$ mentre la metrica β é data da Watt/ $\%utilizzo$. Il loro prodotto risultante é pari a Watt*s che, come ben noto, corrispondono a joule.
2. ΔE_{others} é considerata, in prima approssimazione, trascurabile.
3. La metrica β é un parametro di efficienza energetica intrinseco del processore e pertanto sará univoco per tutte le misure che saranno effettuate.

4. La metrica γ varia al variare dell'applicazione in esecuzione.

Il modello finale su cui si baseranno le misure é dunque:

$$E_{sys} = E_{idle} + \gamma * \beta \quad (2.2)$$

Anticipando quanto verrà spiegato nella sezione dedicata al benchmark energetico, il consumo energetico ad esempio della transazione “Inserimento Nuovo Business Partner” su sistema ERP OFBiz (BP_OFBIZ), può essere stimato con la relazione:

$$E_{BP_{OFBIZ}} = E_{sys_{BP_{OFBIZ}}} - E_{idle} = \gamma_{BP_{OFBIZ}} * \beta \quad (2.3)$$

ovvero, essendo il β fissato per il processore, siamo in grado di stimare il consumo energetico rilevando la sola metrica di utilizzo del processore per l'esecuzione della transazione in esame!

Nei due paragrafi seguenti si illustra la metodologia adottata per la stima della metrica β e il calcolo della metrica γ .

2.2.2 Metrica beta

La metrica β rappresenta il costo energetico di consumo della CPU e mi consente di valutare l'assorbimento di potenza aggiuntivo rispetto allo stato di idle del processore, ovvero quando l'utilizzo é pari a zero. Il beta é dunque una parametro che, in base al suo valore, fornisce un'idea di quanto efficiente sia il processore dal punto di vista energetico. Esempi di possibili comportamenti del processore sono riportati in figura 2.4. La metodologia adottata per stimare il β sulla nostra macchina di test é consistita nell'andare a stressare il processore ad determinate percentuali di utilizzo, rilevandone sperimentalmente i relativi valori di potenza tramite pinza amperometrica. In letteratura esistono numerosi programmi per il testing dell'Hardware e in particolare della CPU (Prime95, Orthos, OCCT, Intel Burn Test solo per citarne alcuni). Questi benchmark sono spesso utilizzati per testare la stabilitá del sistema quando si eseguono interventi di overclocking, underclocking, overvolting e undervolting, ovvero interventi in cui si vanno a modificare la frequenza di clock e il voltaggio della CPU alterandone le performance. L'idea di fondo di questi tests, chiamati anche Torture Tests, é quella di sottoporre la CPU a carichi di lavoro CPU-intensive per lunghi periodi di tempo, al fine di scovare eventuali malfunzionamenti del sistema. Con l'accezione CPU-intensive intendiamo un carico composto da un set di istruzioni che utilizzano l'intero chip, piuttosto che singole unitá. Si ricorda che, nel nostro caso, l'utilizzo di questi programmi non é finalizzato

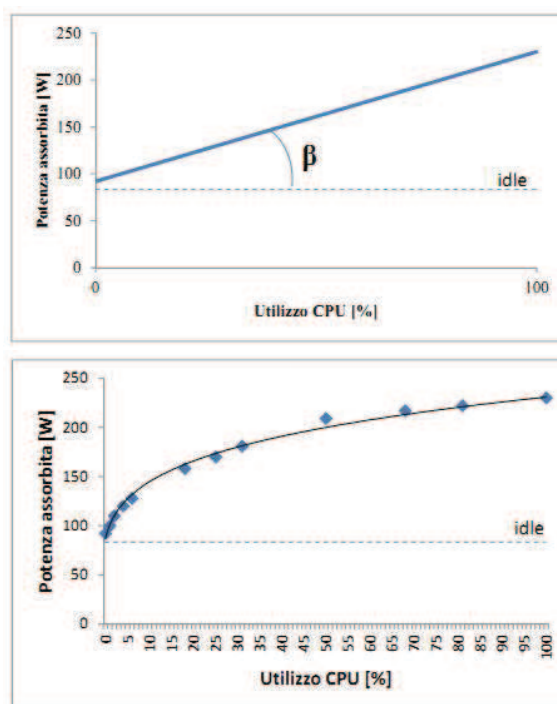


Figura 2.2: Esempi di caratteristica del processore

alla verifica di stabilità del sistema ma ha il solo obiettivo di sollecitare il processore a diversi utilizzi registrando l'assorbimento di corrente.

Il primo programma utilizzato è stato OCCT (OverClock Checking Tool), di cui si mostra l'interfaccia grafica in figura 2.3. Nell'ultima versione (4.1.1), OCCT mantiene inalterate le caratteristiche di semplicità d'uso ma aggiunge interessanti funzionalità. Oltre all'OCCT Stress Test, già conosciuto precedentemente per la bontà del suo algoritmo che massimizza l'utilizzo della FPU (Floating Point Unit) e genera un consistente aumento della temperatura sul chip a causa dell'elevato calore dissipato, è ora disponibile uno stress test basato sulla routine Linpack (OCCT Linpack Test), analogo ad Intel Burn Test. I benchmark Linpack sono utilizzati per valutare le prestazioni nelle operazioni in virgola mobile misurando la rapidità di risoluzione di un sistema di equazioni lineari del tipo $Ax = b$, un problema molto comune in ingegneria. La soluzione in FLOPS, si ottiene tramite eliminazione di Gauss con pivot. L'OCCT Linpack Test ha il vantaggio di stressare maggiormente non solo la FPU, ma anche la memoria cache e l'intero chipset, rappresentando complessivamente un'ottima prova di stabilità per l'intero sistema. In OCCT esistono tre modalità di test (auto, loop infinito, custom) terminato il quale vengono generati automaticamente una serie di grafici che mostrano l'andamento di temperatura e tensione della CPU ed alimentatore.



Figura 2.3: Interfaccia di OCCT

Sebbene un'ottima soluzione per test di stabilità, OCCT non consente di selezionare il numero di core da stressare, impedendoci di rilevare l'assorbimento di potenza a utilizzi diversi dall'idle e dal 100%. Come conseguenza, la caratteristica del processore ottenuta tramite regressione lineare a partire dai risultati di OCCT, risulta dunque un'approssimazione troppo grossolana della curva reale.

Il secondo benchmark considerato è stato Prime95, applicazione realizzata da George Woltman ed utilizzata da GIMPS, un progetto di calcolo distribuito per la ricerca di nuovi numeri primi di Mersenne. Nel corso degli anni, Prime95 è diventato estremamente popolare come utility di test di stabilità. Comprende infatti un torture test, il cui successo è condizione necessaria per effettuare un test di primalità. Il torture test di Prime95 esegue la FFT (Fast Fourier Transform), un algoritmo ottimizzato per calcolare la trasformata discreta di Fourier (DFT) e la sua inversa (IDFT). Si ricorda che la trasformata veloce di Fourier, grazie al basso costo computazionale, è di grande importanza per diverse applicazioni come l'elaborazione di segnali digitali, la soluzione di equazioni differenziali alle derivate parziali o algoritmi per moltiplicare numeri interi di grandi dimensioni. In figura 2.5 si mostra uno screenshot dell'interfaccia di Prime95. Il test può essere configurato modificando il calcolo della FFT in funzione di quali componenti si vogliono testare meglio. Quattro sono le configurazioni possibili:

- Small FFTs
- In-place FFTs
- Blend
- Custom

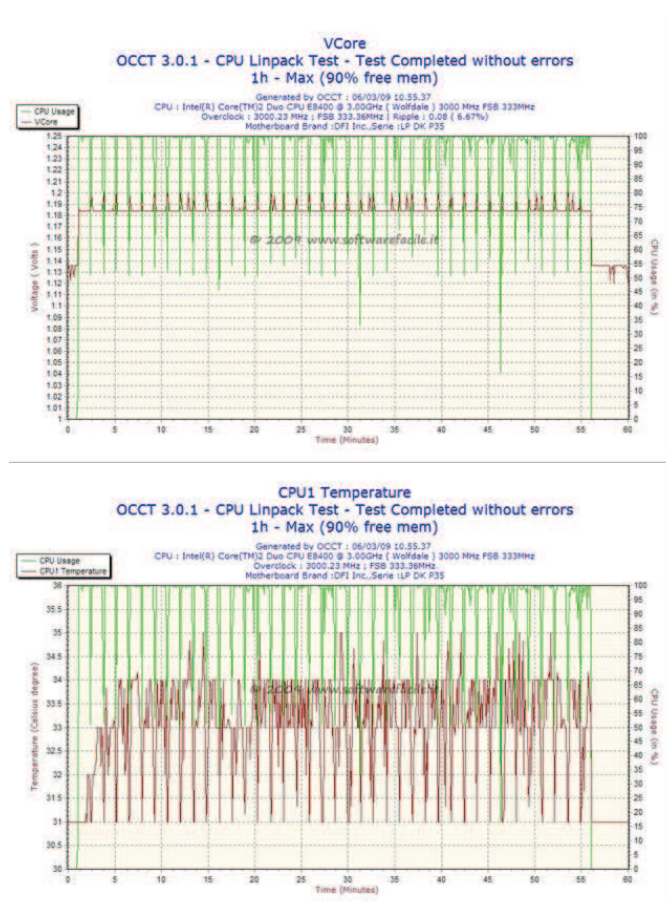


Figura 2.4: Grafici generati da OCCT al termine di un test

Mentre la configurazione Blend, come suggerito dal nome, combina il testing di processore e memoria, le configurazioni Small FFTs e In-place FFTs testano maggiormente la CPU e in particolare la FPU e la memoria cache. Con la configurazione personalizzata invece, è possibile modificare, oltre al livello di utilizzo della CPU e della memoria, anche il numero di thread da eseguire, fino ad un massimo pari al numero di core installati sulla macchina. In questo modo siamo in grado di effettuare fino ad otto rilevamenti della corrente assorbita, dove otto si ricorda è il numero di core della macchina server su cui sono state effettuate le misure. Con Prime95 quindi superiamo ottimamente l'handicap di OCCT che ci fornisce solo un punto sperimentale della curva. Il risultato, come si vedrà nel capitolo successivo, è un' approssimazione più aderente alla curva reale e quindi una stima più precisa della metrica beta.

Ulteriori indagini condotte durante questa fase hanno tuttavia portato a ritenere non soddisfacente neanche la stima della metrica tramite Prime95.

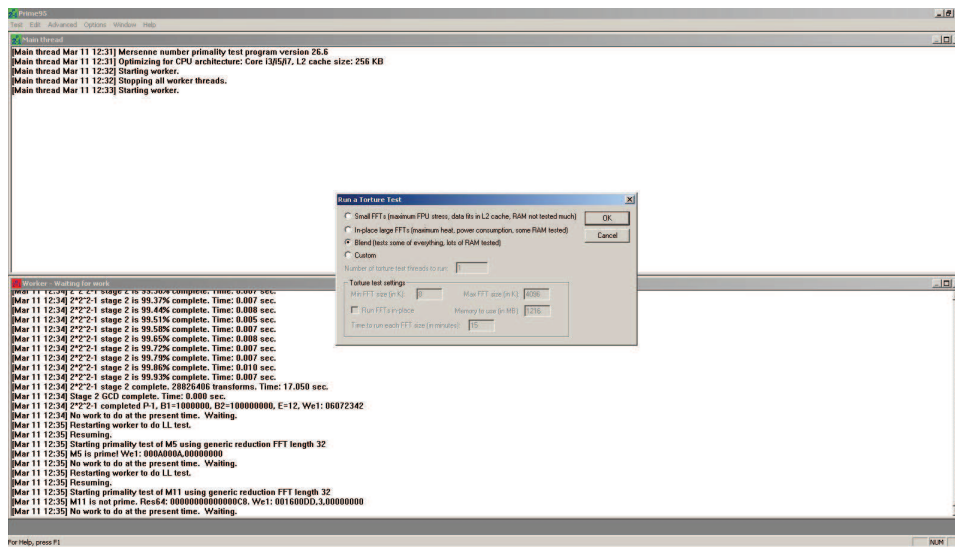


Figura 2.5: Interfaccia di Prime95

Come si vedrà infatti nel capitolo relativo ai risultati sperimentali, il valore del parametro β stimato sia tramite OCCT che Prime95 ha portato a risultati poco soddisfacenti per quanto riguarda la validazione del modello energetico sui workloads dei sistemi ERP di nostro interesse. Ciò è giustificato dalla tipologia di carico a cui viene sottoposto il processore tramite questi benchmark. Si tratta infatti per lo più di carico computazionale oneroso con elevato stress della FPU. Tale carico è ben diverso da quello generato da una transazione di business in cui avvengono maggiormente operazioni di lettura e scrittura su database. Una prima conferma di queste considerazioni si è avuta considerando uno dei tre ERP in esame e generando un carico elevato, tramite LoadRunner, al fine di stressare il processore a diversi utilizzi e ricavare una nuova curva e quindi una nuova stima della metrica beta. Non troppo inaspettatamente, la stima di beta ottenuta è nettamente inferiore alle stime trovate con Prime95 e OCCT e ciò è interpretabile affermando che, a parità di utilizzo di CPU, un carico computazionale oneroso consuma maggiore potenza di un carico computazionale basso. Forti di questa evidenza empirica, al fine di ottenere una stima di beta maggiormente aderente al nostro lavoro si è deciso di implementare una semplice applicazione web che generasse un carico computazionale più simile a quello di un'applicazione di business. Tale applicazione effettua operazioni di lettura e scrittura su database rappresentando per il processore un carico molto più simile a quello relativo alle transazioni dei sistemi ERP oggetto di studio in questo lavoro. Un ulteriore vantaggio dell'applicazione web realizzata è quello di mantenersi indipendenti dai tre sistemi ERP considerati, conservando tuttavia le proprietà che caratterizzano la classe di equivalenza che andremo a

definire per i nostri tre sistemi ERP. Nella sezione dedicata agli strumenti di misura é riportata anche una descrizione architettuale della web application realizzata.

2.2.3 Metrica gamma

La metrica γ abbiamo detto che quantifica l'utilizzo complessivo della cpu quando non é in idle. Se da un lato il nostro obiettivo é quello di stimare il consumo energetico di una singola transazione di business é anche vero che l'esecuzione di una singola transazione comporta un carico sulla cpu pressoché irrilevante e un conseguente gamma prossimo a zero. Per ricavare il consumo energetico di una singola transazione di business si é dunque rilevato il gamma relativo ad un carico significativo di 10 client ottenuto tramite il tool di HP LoadRunner, che verrà presentato piú avanti.

Lo strumento utilizzato per l'acquisizione dei valori di utilizzo é il Performance Monitor di Windows Server 2008, in grado di registrare su file .csv ad un rate di 1 campione al secondo. Una panoramica su questo strumento e su come é stato utilizzato é riportata nei paragrafi successivi. Partendo dai valori di utilizzo, la stima della metrica gamma é stata ottenuto per integrazione numerica, tramite il metodo dei trapezi.

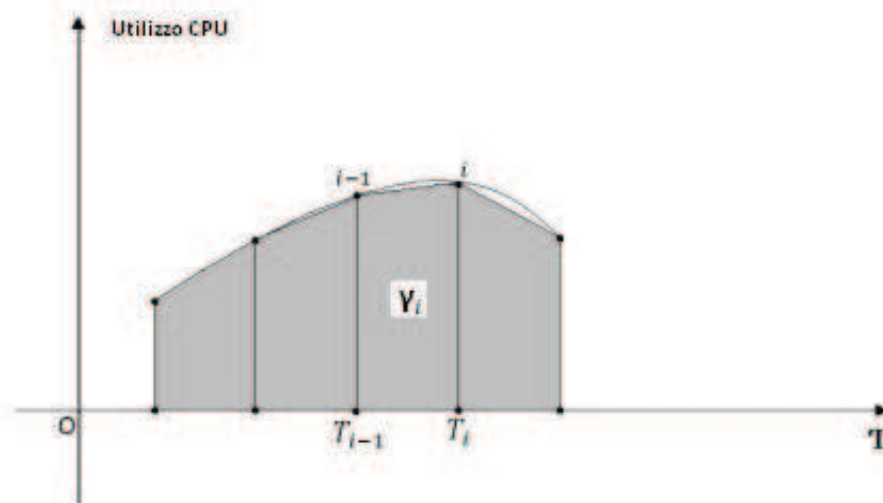


Figura 2.6: Metodo dei Trapezi

2.3 Selezione del Software

Prima di decidere quali applicazioni enterprise considerare, si é reso necessario definire una nuova classe di equivalenza del software, identificando

quali aspetti determinano la comparabilità dal punto di vista energetico tra sistemi software. Si è scelto di considerare una classe di equivalenza caratterizzata da sistemi software:

- ERP (Enterprise Resource Planning)
- Open-Source
- Sviluppati con tecnologie Java
- funzionalmente equivalenti
- che utilizzano DBMS MySQL o PostgreSQL

I sistemi ERP sono applicazioni o molto spesso una suite integrata di applicazioni, deputate alla gestione di tutti gli aspetti di business di un'azienda e di tutte le risorse, attività ed informazioni necessarie al corretto svolgimento delle attività dei processi aziendali. Tipicamente un suite ERP supporta e migliora le performance dei processi di produzione, logistica, distribuzione, il processo di approvvigionamento, la vendita ma anche la contabilità e le risorse umane.

I sistemi di gestione della produzione iniziarono a comparire circa trent'anni fa sotto il nome di sistemi MRP (Manufacturing Requirement Planning). Se inizialmente erano calcolatori di requisiti di materiale, oggi sono in grado di automatizzare e supportare un'intera società. Nel 1990 Gartner coniò l'acronimo ERP (Enterprise Resource Planning) per indicare una suite integrata di applicativi che fanno da supporto non solo alle singole attività operative, ma anche ai processi decisionali, alle attività direzionali e di business intelligence e alla gestione della customer satisfaction.

La suite è in genere organizzata in moduli che possono essere installati secondo una modalità a plug-in, in base alle esigenze del business aziendale. Di seguito sono elencati alcuni moduli tipicamente inclusi in un sistema ERP, accompagnati da una breve descrizione:

Manufacturing Questo modulo include tutto ciò che è relativo alla produzione del prodotto, gestione del flusso di lavoro, controllo qualità, gestione dei costi, processo produttivo, scheduling delle attività.

Supply chain management Modulo che include principalmente inventari, flussi tra ordine e pagamento, ordini, pianificazione della supply chain, e scheduling delle forniture.

Financials Questo modulo include la contabilità generale, analitica e finanziaria.

Projects Modulo che viene utilizzato per la gestione operativa ed economica dei progetti: programmazione attività, pianificazione e consuntivazione di costi/ricavi, budget e controllo disponibilità.

Human resources Modulo che include tutto ciò che riguarda la gestione delle risorse umane. Ciò include pagamenti stipendi, formazione, orari e partecipazioni, e benefit.

Customer relationship management Customer relationship management. Questo modulo include il supporto a vendite e marketing, commissioni, servizi, assistenza, contatti con i clienti e supporto call center o altri canali.

Prima dell'avvento degli ERP, tipicamente ogni dipartimento o business unit aveva il suo sistema informativo. Ciò implicava differenze nell'integrazione dei dati da un sistema all'altro e scarsa sincronizzazione degli stessi, con conseguenti aumenti di tempistiche e latenze operative. Con l'introduzione dei sistemi ERP questi problemi sono oggi molto meno presenti, in quanto il sistema combina dati di applicazioni diverse e tiene tutte le informazioni sincronizzate all'interno della società.

Dopo aver brevemente descritto la classe di sistemi ERP, si presentano ora i sistemi che sono stati scelti per questo lavoro. Come ERP campione si sono utilizzati:

- Openbravo
- Adempiere
- Apache OFBiz

2.3.1 Openbravo

Openbravo é una suite ERP open source sviluppata per piccole e medie imprese dalla Openbravo S.L, una società di software open source commerciale con sede a Pamplona, in Spagna. La suite, basata su interfaccia web, é rilasciata a piccole e medie imprese sotto licenza pubblica Openbravo e la Mozilla Public License che, a differenza della licenza GPL é open-source ma non é libera.

Tramite Openbravo l'utente può monitorare lo stato dell'azienda, e avere informazioni sulla produzione, inventario, custode information, tracciamento degli ordini e workflows interni. Con Openbravo é possibile inoltre creare ed esportare report e dati in diversi formati.

Nato inizialmente come software di gestione amministrativa, Openbravo ERP organizza e centralizza i dati chiave aziendali, facilitando il flusso rapido e semplice delle informazioni lungo tutte le aree coinvolte dai differenti processi aziendali ed evitando duplicazioni e condividendo le informazioni più rilevanti tra le diverse business unit. Le principali funzionalità di Openbravo ERP sono:

- Master Data Management
- Procurement Management
- Warehouse Management
- Project and Service Management
- Production Management
- Sales and CRM
- Financial and Accounting
- Business Intelligence
- Openbravo for Retail

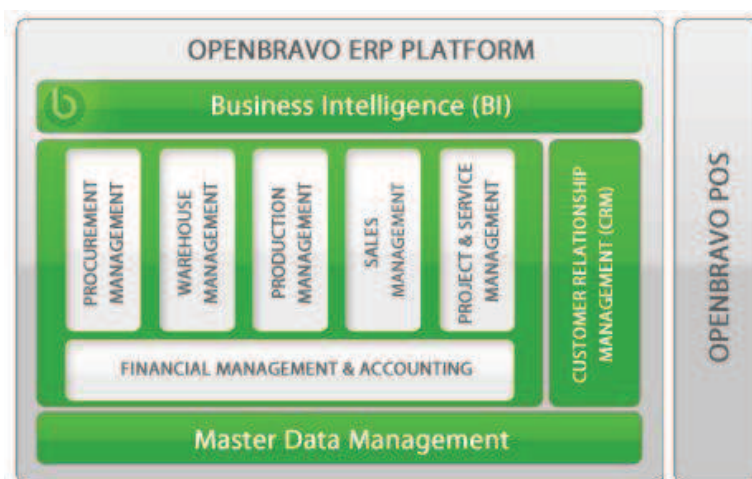


Figura 2.7: Openbravo Platform

La piattaforma di Openbravo é modulare, supporta cioè l'installazione di moduli open-source, relativi ad esempio alla localizzazione o riguardanti estensioni funzionali, connettori per applicazioni di terze parti, funzionalità specifiche per un certo target aziendale e molto altro.

2.3.2 Adempiere

Adempiere é un progetto nato nel settembre 2006 dopo una lunga controversia tra la comunità nata attorno al progetto Compiere e gli sviluppatori, quest'ultimi colpevoli di porre troppa enfasi sulla natura commerciale del progetto e poca sulla natura di condivisione che caratterizza il software open source. É progettato per supportare:

sostanziale al miglioramento della suite, la scalabilità che lo rende adatto sia alle piccole che alle grandi imprese, la modularità, attraverso l'utilizzo di tecnologie basate su interfacce OSGi e la Database Indipendance che, sebbene non ancora pienamente supportata, consente l'utilizzo di diversi DBMS quali PostgreSQL, Oracle e MySQL.

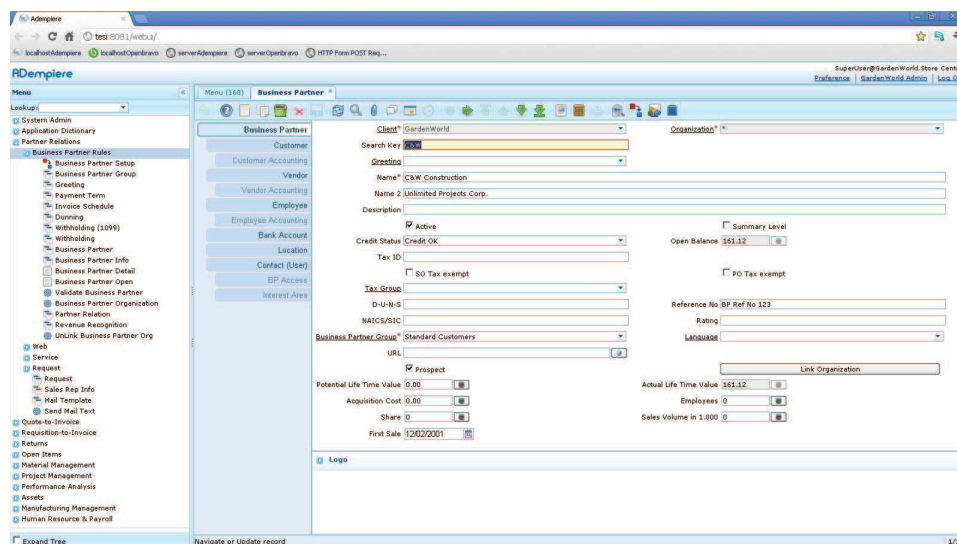


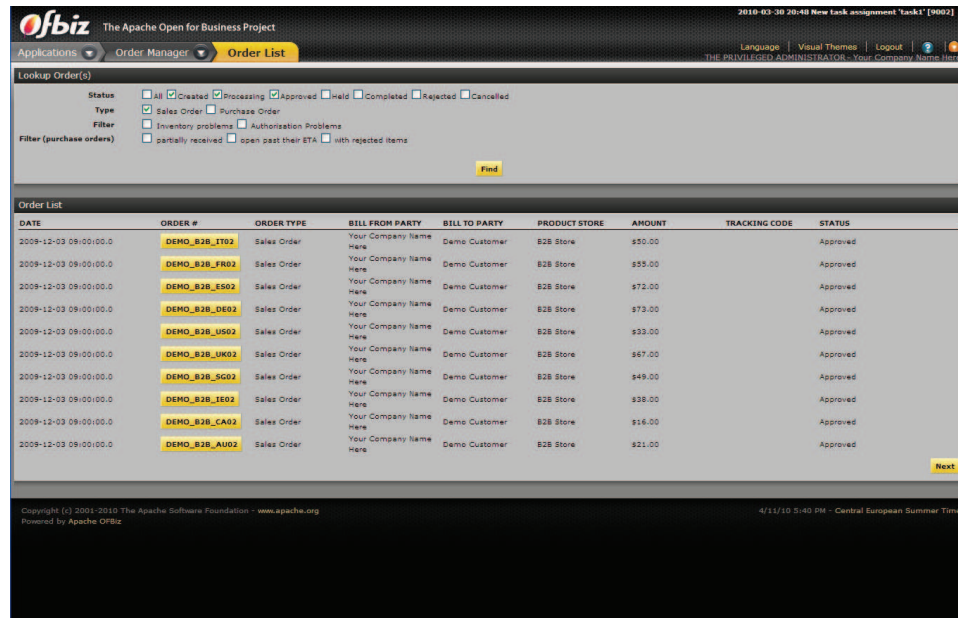
Figura 2.9: Adempiere screenshot

2.3.3 Apache OFBiz

Apache Open for Business (Apache OFBiz) è un progetto della Apache Software Foundation, rilasciato sotto licenza Apache versione 2.0. Tale licenza consente la personalizzazione, l'estensione e modifica del sistema. OFBiz è una suite di applicazioni aziendali basate su un'architettura comune utilizzando dati comuni e componenti a basso accoppiamento secondo una logica di processo, facilitando il riuso dei componenti e l'estensione delle applicazioni con uno sforzo limitato. Apache OFBiz offre un set di funzionalità davvero ampio, tra cui:

- Customer management
- Order management
- Advanced e-commerce
- Catalog management
- Accounting
- Manufacturing management

- Modulo POS (Point of Sales) basato sul framework XUI



The screenshot shows the OFBiz web interface for the Order Manager. At the top, there is a navigation bar with 'Applications', 'Order Manager', and 'Order List'. Below this is a 'Lookup Order(s)' section with filters for Status, Type, and Filter. The main area displays a table of orders.

DATE	ORDER #	ORDER TYPE	BILL FROM PARTY	BILL TO PARTY	PRODUCT STORE	AMOUNT	TRACKING CODE	STATUS
2009-12-03 09:00:00.0	DEMO_B2B_IT02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$50.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_FR02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$55.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_ES02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$72.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_DE02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$73.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_US02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$33.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_UK02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$67.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_SG02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$49.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_IE02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$38.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_CA02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$16.00		Approved
2009-12-03 09:00:00.0	DEMO_B2B_AU02	Sales Order	Your Company Name Here	Demo Customer	B2B Store	\$21.00		Approved

Figura 2.10: OFBiz screenshot

Le soluzioni presentate sono sviluppate in linguaggio Java e sono basate principalmente su interfaccia web. Inoltre utilizzano PostgreSQL come RDBMS. Questo implica una piena comparabilità nelle analisi che andremo a svolgere nel seguito. Infine, come già detto, sono soluzioni open source, il che significa aver la possibilità di svolgere ulteriori analisi anche a livello di codice e non solo a livello funzionale.

2.4 Definizione benchmark energetico

Ora che abbiamo introdotto i sistemi ERP su cui andremo a condurre la nostra analisi, si prosegue con la descrizione di ciò che andremo effettivamente a misurare. L'obiettivo che si è tentato di raggiungere in questa fase, è la definizione di una specifica di benchmark energetico, prendendo spunto dalla specifica TPC-C, descritta in appendice.

La presentazione inizia con la descrizione dettagliata di quattro transazioni di business, attività chiamata anche profilazione. Una transazione di business è una sequenza di operazioni che l'utente effettua sul sistema ERP. Le transazioni qui presentate sono il risultato di un'attenta analisi funzionale dei tre sistemi ERP in esame, con lo scopo di identificare le operatività comuni e i dati condivisi per il loro completamento. La fase successiva alla profilazione è stata la definizione di workflow o scenari tipici, la cui implementazione è stata realizzata tramite scripts della suite HP LoadRunner. Una descrizione approfondita di questo strumento professionale segue nei paragrafi successivi.

La definizione dell'architettura del sistema testato (SUT) e l'implementazione degli scenari tramite script LoadRunner, costituiscono a tutti gli effetti l'implementazione del benchmark proposto.

2.4.1 Profilazione Transazioni

Come si è detto, il primo step è consistito nell'analizzare dal punto di vista funzionale i tre sistemi ERP scelti. Questa fase ha portato all'identificazione del seguente set di transazioni:

1. Inserimento di un nuovo Business Partner
2. Inserimento di un nuovo Prodotto
3. Inserimento di un nuovo Ordine di Acquisto
4. Inserimento di un nuovo Ordine di Vendita

che verranno ora descritte.

Inserimento di un nuovo Business Partner

Un business partner é un'entitá commerciale legata ad un'organizzazione da un rapporto di collaborazione continuativa per quanto riguarda la distribuzione, promozione e vendita di beni e servizi. Un business partner puó essere un fornitore di materiali, servizi, risorse finanziarie oppure un consumatore, che stipula uno o piú contratti di acquisto di beni o servizi con l'azienda, oppure un membro del personale. Le tipologie di business partner (Vendor, Customer, Employee) non sono in mutua esclusione. La transazione prevede l'inserimento nel sistema di informazioni relative al nuovo partner commerciale, le quali sono utilizzate dal sistema per la generazione di tutta la documentazione associata.

Profilo

1. Creazione Business Partner Group (BPG)

Un Business Partner Group caratterizza BP simili dal punto di vista contabile

- 1.1. Inserimento informazioni BPG (nome, descrizione, search key)
- 1.2. Definizione piani di Accounting BPG (tutti i business partner afferenti li ereditano)

2. Creazione Greetings

Definizione dei titoli associati ad un contatto o ad un Business Partner

- 2.1. Inserimento informazioni Greeting (id greeting, nome greeting)

3. Creazione termini di pagamento (Payment Terms)

Un Payment Term definisce le condizioni di pagamento delle fatture. Criteri di pagamento diversificati possono essere definiti attraverso l'uso delle Lines

- 3.1. Inserimento Informazioni payment term (id,search key, descrizione, Fixed due date, next business day)
- 3.2. Definizione Schedules/Lines associati al payment term (opzionale)

4. Creazione piani di fatturazione (Invoice Schedules)

Una piano di fatturazione definisce la frequenza di emissione delle fatture al Business Partner

- 4.1. Inserimento informazioni piano (id piano, descrizione)
- 4.2. Inserimento soglia minima per invio fattura
- 4.3. Selezione frequenza di fatturazione (giornaliera, settimanale, mensile, bimestrale)

- 4.4. Selezione giorno del mese per generazione fattura
- 4.5. Selezione invoice day cut-off (oltre il quale non posso modificare la fattura)

5. Creazione Business Partner (BP)

5.1. Creazione BP Header (obbligatorio)

- 5.1.1. Inserimento informazioni BP (nome, search key, descrizione, URL, TAXID)
- 5.1.2. Selezione greeting
- 5.1.3. Selezione BPG
- 5.1.4. Abilitazione esenzione tasse

5.2. Creazione profili Customer (opzionale)

- 5.2.1. Selezione piano di fatturazione
- 5.2.2. Selezione metodo di fatturazione (After delivery, immediate..)
- 5.2.3. Selezione piano di consegna (After Receipt, Availability, Complete Line, Complete Order..)
- 5.2.4. Selezione metodo di consegna (Delivery, Shipper, Pickup)
- 5.2.5. Selezione Price List
- 5.2.6. Selezione metodo di pagamento (Cash, On Credit, Credit Card ...)
- 5.2.7. Selezione termini di pagamento
- 5.2.8. Selezione Responsabile Vendite
- 5.2.9. Inserimento monte crediti
- 5.2.10. Definizione Customer Accounting (default quello associato al BPG afferente)

5.3. Creazione profili Vendor (opzionale)

- 5.3.1. Selezione metodo di pagamento acquisti
- 5.3.2. Selezione termini di pagamento acquisti
- 5.3.3. Selezione Purchase Price List
- 5.3.4. Definizione Vendor Accounting (default quello associato al BPG afferente)

5.4. Creazione profili Employee (opzionale)

- 5.4.1. Abilitazione BP come Employee
- 5.4.2. Definizione Employee Accounting (default quello associato al BPG afferente)

5.5. Creazione Locations (opzionale)

- 5.5.1. Inserimento informazioni location (id location, telefono, fax)

- 5.5.2. Inserimento Address (paese, regione, città, via, CAP)
- 5.5.3. Abilitazione location per spedizione merci
- 5.5.4. Abilitazione location per invio fatture
- 5.6. **Creazione contatti (opzionale)**
 - 5.6.1. Inserimento informazioni contatto (nome, descrizione, commenti, posizione, mail, telefono)
- 5.7. **Creazione profili bancari (opzionale)**

Inserimento di un nuovo Prodotto

Un prodotto rappresenta un bene acquistato o prodotto oppure una risorsa o servizio che viene offerto. I prodotti sono raggruppati in categorie di prodotto, le quali sono caratterizzate da profili di accounting specifici. Le quantità di prodotto sono definite secondo una o più unità di misura, mentre informazioni relative alle imposte sono raggruppate in tax categories.

Profilo

1. **Creazione Categoria Prodotto**
 - 1.1. Inserimento informazioni Categoria Prodotto (nome, search key, descrizione)
 - 1.2. Selezione Asset Category
 - 1.3. Inserimento Planned Margin
2. **Creazione Unit of Measure (UOM)**
 - 2.1. Inserimento informazioni UOM (Codice UOM, simbolo, nome, descrizione)
 - 2.2. Inserimento Standard Precision (arrotondamento)
 - 2.3. Inserimento Costing Precision (arrotondamento)
3. **Creazione Tax Category**
 - 3.1. Inserimento Tax Category (nome, descrizione)
4. **Creazione Prodotto**
 - 4.1. **Creazione Product Header (obbligatorio)**
 - 4.1.1. Inserimento informazioni Prodotto (search key, nome, descrizione)
 - 4.1.2. Selezione product category
 - 4.1.3. Selezione UOM
 - 4.1.4. Selezione Tax Category

- 4.1.5. Selezione Product Type
- 4.1.6. Inserimento peso
- 4.1.7. Inserimento volume
- 4.1.8. Attivazione/Disattivazione Purchased checkbox
- 4.1.9. Attivazione/Disattivazione Sold checkbox

4.2. Definizione profili di pricing

- 4.2.1. Selezione listino prezzi
- 4.2.2. Inserimento prezzo di listino
- 4.2.3. Inserimento prezzo standard
- 4.2.4. Inserimento limite prezzo

Inserimento di un nuovo Ordine di Acquisto

Un ordine di acquisto é un documento contenente le informazioni relative a prodotti e/o servizi richiesti ad un fornitore. L'ordine viene evaso previa accordi contrattuali che definiscono i termini di fatturazione, di consegna e pagamento. Il documento contiene la specifica dell'ordine in termini di order lines.

Profilo

1. Creazione ordine di acquisto

- 1.1. Definizione intestazione ordine (riferimento ordine, descrizione ordine, data ordine, indirizzo di consegna, indirizzo di fatturazione, contatto)
- 1.2. Selezione magazzino di consegna
- 1.3. Selezione listino prezzi
- 1.4. Selezione metodo di pagamento
- 1.5. Inserimento riferimenti
- 1.6. Selezione stato ordine
- 1.7. Salvataggio

2. Creazione order lines

- 2.1. Creazione nuova order line
- 2.2. Caricamento riferimento ordine
- 2.3. Inserimento data
- 2.4. Selezione prodotto
- 2.5. Inserimento quantità
- 2.6. Inserimento prezzo unitario

- 2.7. Selezione imposta
- 2.8. Selezione sconto
- 2.9. Salvataggio

Inserimento di un nuovo Ordine di Vendita

Un ordine di vendita é un documento contenente le informazioni relative a prodotti e/o servizi richiesti da un cliente. L'ordine é evaso previa accordi contrattuali che definiscono i termini di fatturazione, di consegna e pagamento. Il documento contiene la specifica dell'ordine in termini di order lines.

Profilo

1. Creazione ordine di vendita

- 1.1. Definizione intestazione ordine (riferimento ordine, descrizione ordine, data ordine, indirizzo di consegna, indirizzo di fatturazione, contatto)
- 1.2. Selezione magazzino di consegna
- 1.3. Selezione listino prezzi
- 1.4. Selezione rappresentante vendita
- 1.5. Inserimento riferimenti
- 1.6. Selezione stato ordine
- 1.7. Salvataggio

2. Creazione order lines

- 2.1. Creazione nuova order line
- 2.2. Caricamento riferimento ordine
- 2.3. Selezione prodotto
- 2.4. Inserimento quantità
- 2.5. Inserimento prezzo unitario
- 2.6. Selezione imposta
- 2.7. Selezione sconto
- 2.8. Salvataggio

2.4.2 Definizione Scenari

Il secondo step nella definizione del nostro benchmark é l'identificazione di un set di scenari campione che rappresentino un utilizzo tipico delle transazioni appena definite. Questo step é fondamentale in quanto ci permette di caratterizzare le misure che andremo a fare successivamente con gli strumenti di misura. Di seguito si riportano, per ciascuna delle transazioni di business precedentemente descritte, il relativo scenario che si é deciso di considerare e misurare.

Scenario Inserimento Nuovo Business Partner

1. Creazione 1 Business Partner Group
2. Creazione 1 Payment Term
3. Creazione 1 Invoice Schedule
4. Creazione 1 Business Partner con ruolo Customer

Scenario inserimento nuovo Prodotto

1. Creazione 1 categoria prodotto
2. Creazione 1 UOM
3. Creazione 1 Tax Category
4. Creazione 1 prodotto

Scenario Inserimento Nuovo Ordine di Acquisto

1. Creazione 1 ordine di acquisto
2. Creazione 2 order lines

Scenario Inserimento nuovo Ordine di Vendita

1. Creazione 1 ordine di vendita
2. Creazione 2 order lines

Abbiamo quindi completato la specifica del nostro benchmark la cui implementazione servirá per validare il modello energetico presentato nelle sezioni precedenti. Segue ora la descrizione della suite LoadRunner e di come é stata utilizzata per la creazione degli script delle transazioni.

2.4.3 La suite HP LoadRunner

Le moderne applicazioni di oggi sono caratterizzate da un elevato grado di complessità e spesso risultano difficili da testare. Il testing di tipo single-user risulta per lo più adatto a verificare la correttezza funzionale o aspetti di usabilità. Ma che dire circa le performance? Un tipico scenario di test potrebbe esser rappresentato ad esempio da 1000 utenti che effettuano il login contemporaneamente o in un range temporale molto limitato. Qual é il tempo di risposta del sistema sotto un tale carico? Qual é il limite alla scalabilità oltre il quale il sistema crasha o diventa praticamente inutilizzabile?

É chiaro che metodi di testing tradizionali e manuali risultano inefficaci per realizzare scenari di test come quello appena descritto.

HP Load Runner é uno strumento per il performance testing automatizzato delle applicazioni realizzato da Hewlett-Packard. É in grado di stressare l'intera infrastruttura, in gergo SUT (System Under Test), emulando migliaia di utenti e utilizzando monitor di performance per esaminare il comportamento del sistema, identificando ed isolando eventuali colli di bottiglia che si possono verificare negli apparati di rete, piuttosto che causati dalla saturazione della capacità elaborativa del sistema.

I vantaggi principali derivanti dal suo utilizzo sono:

- Replacement degli utenti con virtual users che simulano il comportamento di un utente reale;
- Riduzione significativa dell'hardware richiesto per effettuare l'attività di testing. Su ogni macchina fisica possono essere eseguiti numerosi virtual users;
- Single point of Control dei virtual users tramite il Controller;
- Monitoraggio delle performance online, consentendo il tuning del sistema durante la fase di testing;
- Raccolta automatica e reporting dei dati di performance;
- Ripetibilità dei test, essendo questi completamente automatizzati.

La suite si compone di tre strumenti principali:

- Virtual User Generator (VuGen): LoadRunner sostituisce l'utente umano con un utente virtuale (Virtual User o Vuser) per la generazione del carico. Quando viene eseguito uno scenario di test, ogni Vuser emula le azioni di un utente reale nella interazione con l'applicazione in esame. Più precisamente ogni Vuser esegue uno Vuser script. Il VuGen é il tool che consente la creazione dello Vuser script.

- Controller: serve per la definizione di uno scenario di test, a partire da uno o piú Vuser scripts. Questo tool consente di controllare i Vusers durante lo scenario da una singola workstation.
- Performance Analysis: durante l'esecuzione di uno scenario, Load Runner colleziona molteplici informazioni relative per esempio alle performance della rete piuttosto che relative alle risorse del server. Tali informazioni sono raccolte ed elaborate dal Performance Analysis da cui é possibile generare utili report e grafici.

L'attività di testing di una web application con LoadRunner puó essere descritta attraverso un processo di testing composto da diverse fasi, come illustrato in figura 2.11



Figura 2.11: Processo di Testing

Step 1: Planning the Test

Una buona pianificazione di test rappresenta la chiave di successo. Un piano di test ben definito assicura la realizzazione di scenari in linea con gli obiettivi di test prefissati. In figura 2.12 si mostra il processo di pianificazione, a sua volta articolato in tre steps.

Il primo passo é l'analisi dell'applicazione, ovvero l'identificazione dei componenti hardware e software che fanno parte del SUT. L'obiettivo di questa analisi é la progettazione di un ambiente di test, con LoadRunner, il piú aderente possibile con il SUT. La descrizione della configurazione del sistema puó essere accompagnata da uno schema dell'architettura del sistema, corredato di informazioni riguardanti sia le risorse utilizzate, sia il modello



Figura 2.12: Planning del Processo di Test

di utilizzo dell'applicazione. Di seguito si mostra, a titolo esemplificativo, un possibile schema risultante dall'analisi.

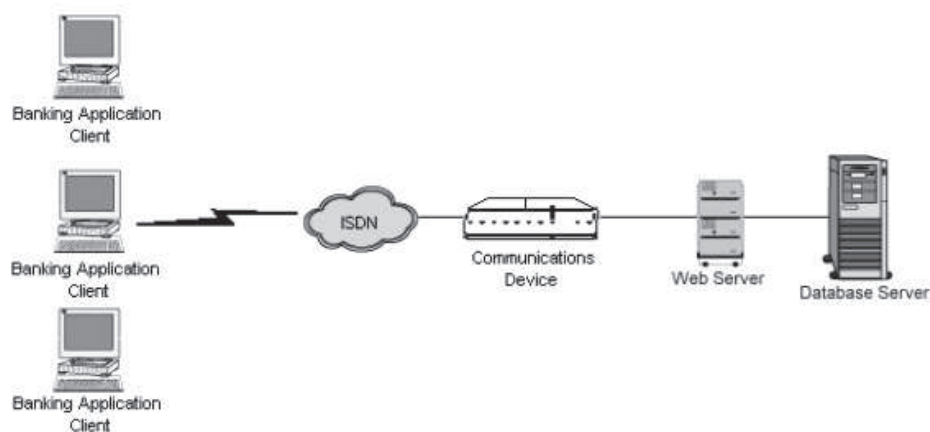


Figura 2.13: Esempio di architettura del SUT

Il secondo passo é la definizione degli obiettivi di test che possono essere:

- Ridurre il tempo di risposta
- Identificare la configurazione hardware ottimale
- Controllare l'affidabilità del sistema
- Misurare la system capacity
- Identificare colli di bottiglia

Il terzo ed ultimo passo della pianificazione consiste nel decidere come usare LoadRunner al fine di raggiungere gli obiettivi definiti. In particolare tipiche decisioni a questo livello sono:

- Definire lo scope delle misure di performance: in quali punti nel ciclo di vita dell'applicazione é opportuno effettuare il test
- Definire i task dei Vuser: ovvero identificare un uso tipico della web application in esame.
- Decidere il numero di Vusers sulla base di un modello di utilizzo significativo per l'applicazione
- Scegliere l'infrastruttura hardware e software di test.

Step 2: Creating Vuser Scripts

Come detto, uno Vuser emula l'utente reale nell'interazione con la web application. Un "Vuser script" contiene le azioni eseguite dallo Vuser durante l'esecuzione dello scenario. Più precisamente nello Vuser script sono contenuti i task che saranno:

- Eseguiti da ciascun Vuser
- Eseguiti simultaneamente da molteplici Vusers
- Misurati come transazioni

Step 3: Creating the Scenario

Lo scenario descrive completamente una sessione di testing. Include:

- l'elenco dei Load Generator, ovvero le macchine su cui vengono lanciati i Vusers
- l'elenco degli script che devono essere eseguiti nello scenario
- il numero di Vusers che eseguono lo scenario, ovvero il carico
- il profilo di carico nel tempo

HP LoadRunner Controller consente la definizione di due tipologie di scenario:

- Manual: l'utente definisce gli script che devono essere eseguiti, il numero (o la percentuale) di Vuser per ciascuno script e i Load Generators per la loro esecuzione.
- Goal-oriented: lo scenario viene automaticamente generato sulla base degli obiettivi di test specificati

Step 4: Running the Scenario

Durante l'esecuzione dello scenario, LoadRunner misura e registra le transazioni eventualmente definite in ogni Vuser scripts. É inoltre possibile monitorare le prestazioni del sistema online.

Step 5: Monitoring the Scenario

É possibile monitorare l'esecuzione dello scenario tramite monitors. Alcune delle risorse monitorabili sono:

- Le transazioni
- Risorse di sistema
- Web/application server
- Database server
- Interfaccia di rete

Step 6: Analyzing Test Results

Tramite grafici e report autogenerati da LoadRunner é possibile analizzare le performance del sistema registrate durante la sessione di test.

Per gli scopi di questo lavoro il processo di testing seguito ha coperto i soli step di pianificazione, creazione degli script, creazione degli scenari e la loro esecuzione. In particolare, per ogni transazione del nostro benchmark energetico sono state effettuate le seguenti attività:

1. registrazione della sessione web con Vuser script Generator;
2. correlazione dello script;
3. parametrizzazione dello script
4. creazione dello scenario di test con LoadRunnerController;
5. esecuzione dello scenario di test.

La correlazione dello script é risultata essere la fase piú critica in quanto si é reso necessario individuare tutti i parametri scambiati tra il browser e il server al fine di rendere la sessione unica e quindi replicabile.

Per quanto riguarda infine la parametrizzazione, si é proceduto a identificare e parametrizzare tutte le informazioni salvate su database come chiave primaria. Di seguito si mostra una porzione di codice implementata per la correlazione di un parametro interno usato da Adempiere ERP.

```

*****
...
web_cache_cleanup ();
web_cleanup_cookies ();
web_set_max_html_param_len (" 1024");

web_reg_save_param (" clientInfo ",
    "LB=zk.process (' clientInfo ', \"",
    "RB=\\");",
    "NotFound=ERROR",
    LAST);

web_url (" index.zul",
    "URL=http://192.168.137.1:8080/webui/index.zul",
    "TargetFrame=",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=",
    "Snapshot=t2.inf",
    "Mode=HTML",
    EXTRARES,
    "Url=theme/default/images/login-box-header.png",
    "Url=theme/default/images/login-box-bg.png",
    "Url=theme/default/images/login-box-footer.png",
    ...
    "Url=images/Zoom16.png", ENDITEM,
    "Url=theme/default/images/adtab-left-bg.png",
    "Url=images/Editor16.png", ENDITEM,
    LAST);
...
*****

```

2.5 Strumenti di misura

2.5.1 Pinza Amperometrica e Multimetro Digitale

Le misurazioni di consumo energetico sono state prese direttamente sul cavo di alimentazione del server, tramite una pinza amperometrica C113 Cauvin-Arnaux. Una pinza amperometrica é un dispositivo elettrico con due ganasce che, aprendosi, consentono di circondare un conduttore elettrico. Ciò permette la misurazione della corrente elettrica per effetto Hall, senza avere contatti fisici con il conduttore, o disconnetterlo per permettere



Figura 2.14: Pinza Amperometrica

l'inserimento in serie di un amperometro o di una sonda. Il vantaggio di questo strumento é l'acquisizione della misura senza interferire con la misura stessa.

La corrente misurata dalla pinza é letta da un multimetro digitale, modello Extech 383274, collegato attraverso una porta parallela ad un pc secondario su cui sono stati archiviati ed elaborati i dati acquisiti. Il multimetro, infatti, é corredato di una semplice utility per la visualizzazione e memorizzazione su disco dei valori di corrente letti alla frequenza massima di 1Hz.

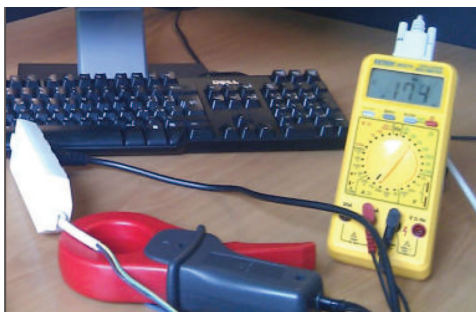


Figura 2.15: Multimetro digitale Extech 383274

La pinza amperometrica e il multimetro digitale sono stati utilizzati in questo lavoro per due finalità:

- identificare la curva caratteristica del processore per determinare la metrica β ;
- determinare l'energia per ogni misura al fine di validare il nostro modello energetico stimandone l'errore percentuale;

La macchina di test, su cui sono state condotte tutte le misure, é un server IBM x3500 con le seguenti specifiche tecniche:

- 2 processori Intel Xeon Quad-core x5450 3.00 GHz (12 MB L2 cache), FSB a 1333 MHz;
- Memoria di sistema 17GB PC2-5300 DDR2 SDRAM;
- Sistema Operativo Windows Server 2008.

2.5.2 Windows Performance Monitor

Il Performance Monitor di Windows é uno snap-in¹ MMC (Microsoft Management Console) che offre strumenti per l'analisi delle prestazioni del sistema. Da una sola console é possibile monitorare le prestazioni dell'applicazione e dell'hardware in tempo reale, personalizzare i dati che si desiderano raccogliere nei registri, definire azioni automatiche, generare rapporti e visualizzare i dati delle performance precedenti in diversi modi. Il monito-

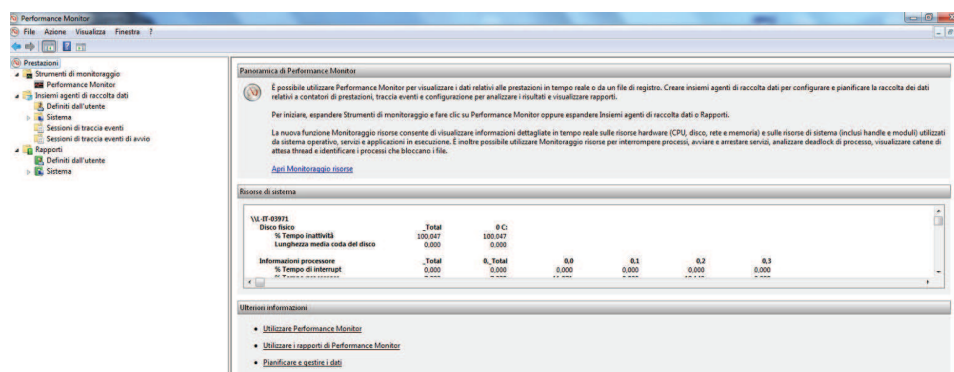


Figura 2.16: Performance Monitor di Windows Server 2008

raggio delle performance avviene tramite i performance counters, che sono indicatori che misurano lo stato del sistema. É possibile includerli nel sistema operativo o possono far parte delle singole applicazioni. Performance Monitor di Windows richiede il valore corrente dei contatori delle prestazioni a intervalli di tempo specifici.

Un altro componente fondamentale utilizzato in questo lavoro é l'insieme agenti di raccolta dati, componente di monitoraggio e segnalazione delle prestazioni in Performance Monitor di Windows. Vengono organizzati piú punti di raccolta di dati in un solo componente che possono essere utilizzati per esaminare o registrare le prestazioni. Un insieme agenti di raccolta dati puó essere creato e quindi registrato singolarmente, raggruppato con un altro insieme agenti di raccolta dati e incorporato nei registri, visualizzato in Monitoraggio prestazioni, configurato per generare avvisi quando vengono raggiunte le soglie oppure utilizzato da altre applicazioni non Microsoft.

¹Uno snap-in é il componente di base di una console. Gli snap-in possono essere utilizzati solo all'interno delle console e non possono essere eseguiti in modo indipendente.

Può essere associato alle regole di pianificazione per la raccolta di dati in periodi di tempo specifici. È possibile configurare l'esecuzione delle attività di Windows Management Interface (WMI) al completamento della raccolta dell'insieme agenti di raccolta dati. In Insiemi agenti di raccolta dati sono contenuti i tipi seguenti di agenti di raccolta dati:

- Contatori prestazioni
- Dati di analisi eventi
- Informazioni relative alla configurazione del sistema (valori chiave del Registro di sistema)

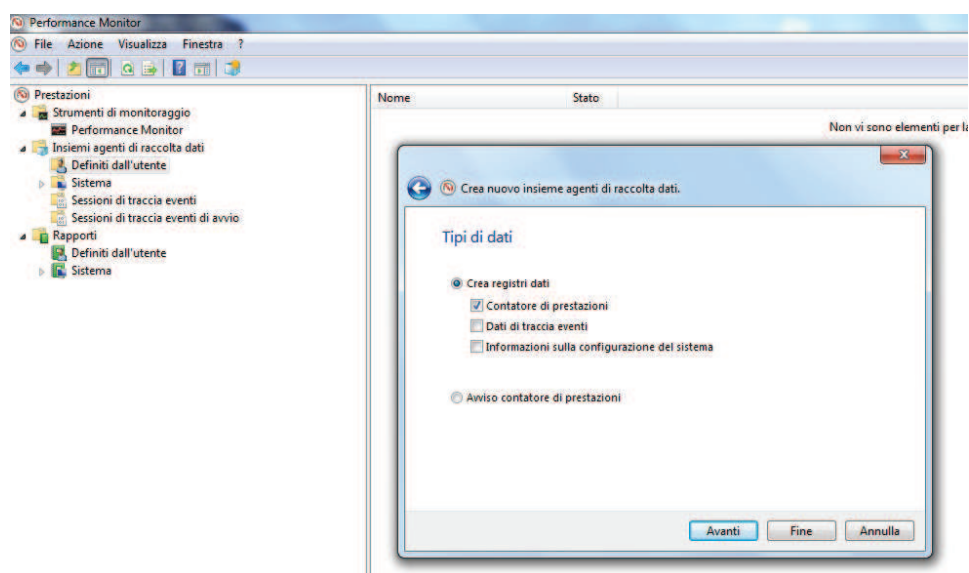


Figura 2.17: Performance Monitor: creazione del Set Agent

Per i nostri scopi, ovvero monitorare l'utilizzo del processore durante l'esecuzione degli script, sono stati effettuati i seguenti passi:

- creazione di un nuovo insieme agenti di raccolta dati
- inserimento di un agente di raccolta di tipo contatori di performance
- selezione contatore
- inserimento path di salvataggio su disco dell'insieme agenti
- configurazione formato .csv per il file registro del contatore
- schedulazione avvio e arresto dell'agente creato

2.5.3 CPU Stress Web Application

Al fine di stimare correttamente la metrica β si é deciso di implementare un'applicazione web che simula il processo di Procurement di un'azienda manifatturiera e che quindi, dal punto di vista del carico computazione sul processore, si avvicina alle principali transazioni eseguiti su un sistema ERP. Si noti che, essendo l'obiettivo di questa applicazione quello di simulare una tipologia di carico orientata ai sistemi di business, ci si é accontentati di implementare una versione del processo assai semplificata rispetto ad un caso reale, mantenendo tuttavia un flusso di esecuzione dell'applicazione in linea con le principali attività del processo di approvvigionamento, di cui si mostra lo schema in figura 2.18.

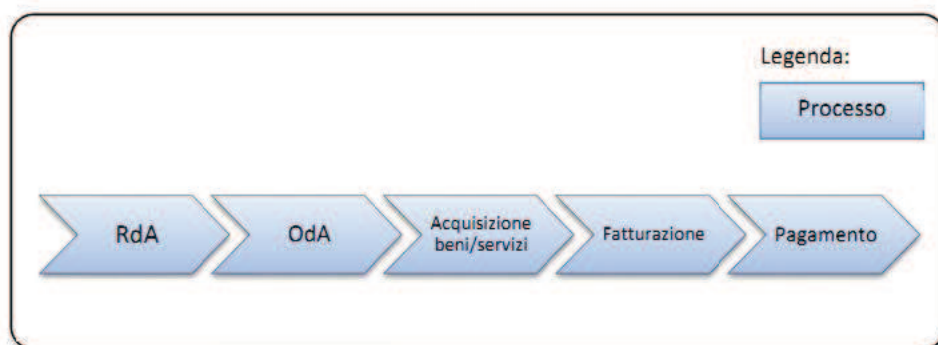


Figura 2.18: Processo di eProcurement

Le attività principali del processo sono:

Richiesta di acquisto : dall'analisi del fabbisogno di beni/servizi, si procede all'emissione della richiesta di acquisto del prodotto/servizio necessario.

Ordine di acquisto : a fronte della richiesta di acquisto, l'ufficio acquisti emette un ordine che viene inviato al fornitore. L'ordine di acquisto può essere riferito a un accordo contrattuale o meno.

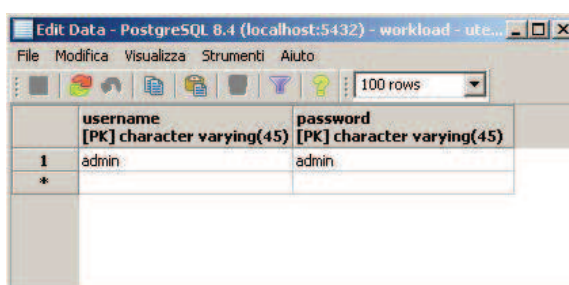
Acquisizione beni/servizi : in questa fase si registra l'acquisizione del bene o l'avanzamento dell'erogazione del servizio.

Fatturazione : processo durante il quale viene effettuata la registrazione della fattura e la verifica della correttezza di quanto emesso dal fornitore rispetto ai beni consegnati/servizi erogati.

Pagamento : tutto ciò che concerne il pagamento derivante dalle operazioni di approvvigionamento di beni/servizi.

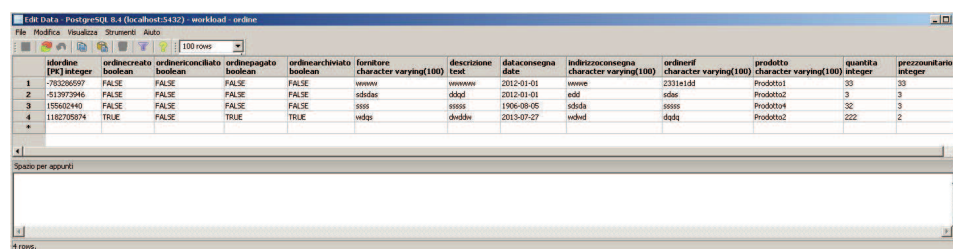
La web application realizzata si appoggia ad un semplice database costituito da due tabelle:

- tabella UTENTE: per la gestione dell'accesso previa autenticazione
- tabella ORDINE: per la gestione del ciclo di vita dell'ordine, dalla richiesta di acquisto alla sua archiviazione



	username [PK] character varying(45)	password [PK] character varying(45)
1	admin	admin
*		

Figura 2.19: Tabella Utenti creata per il controllo d'accesso



idordine [PK] integer	ordinecreato boolean	ordineconclutato boolean	ordinepagato boolean	ordinearchiviato boolean	fornitore character varying(100)	descrizione text	dataconsegna date	indirizzoconsegna character varying(100)	ordineid character varying(100)	prodotto character varying(100)	quantita integer	prezzounitario integer
1	-783266597	FALSE	FALSE	FALSE	wwwwww	wwwwww	2012-01-01	wwwwe	2331e1d1	Prodotto1	33	33
2	-513973966	FALSE	FALSE	FALSE	sdsdas	dsksd	2012-01-01	eddi	sds	Prodotto2	3	3
3	115562140	FALSE	FALSE	FALSE	ssss	ssss	1106-09-05	sdsds	sssss	Prodotto1	32	3
4	1182705974	TRUE	FALSE	TRUE	wedqs	dwddn	2013-07-27	weddd	dqds	Prodotto2	222	2
*												

Figura 2.20: Tabella Ordine per la gestione delle informazioni di un ordine

La tabella ORDINE contiene tutti i dati associati ad un ordine (OdA). In particolare all'atto della creazione di una richiesta di acquisto (RdA) sono vengono scritte su database le seguenti informazioni:

- riferimento ordine
- descrizione dell'ordine
- prodotto ordinato
- fornitore
- quantità
- prezzo unitario

La gestione di un ordine viene effettuata tramite flags i cui valori determinano lo stato dell'OdA all'interno del suo ciclo di vita.

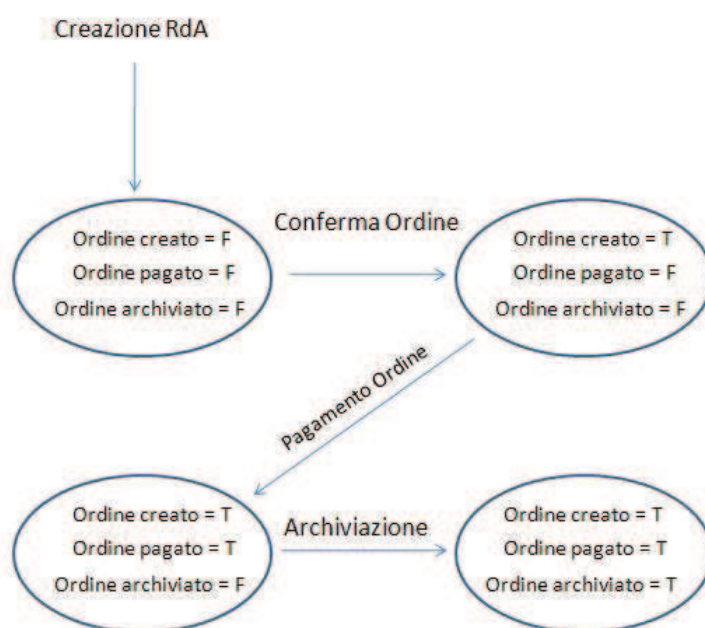


Figura 2.21: Ciclo di Vita dell'Ordine

Struttura dell'applicazione

L'applicazione realizzata é una semplice web application java costituita da 6 pagine jsp rispettivamente per la gestione di:

- autenticazione dell'utente
- homepage dell'applicazione
- inserimento nuova RdA
- elenco RdA
- elenco OdA
- archivio

L'elaborazione di queste pagine viene eseguita da un set di servlet che si occupano essenzialmente di leggere/scrivere i dati da/su database, elaborarli e restituirli al browser.

A titolo esemplificativo si mostra la sequenza di screenshot relativa all'intero ciclo di vita di un ordine, dalla creazione della RdA fino all'archiviazione dell'OdA.

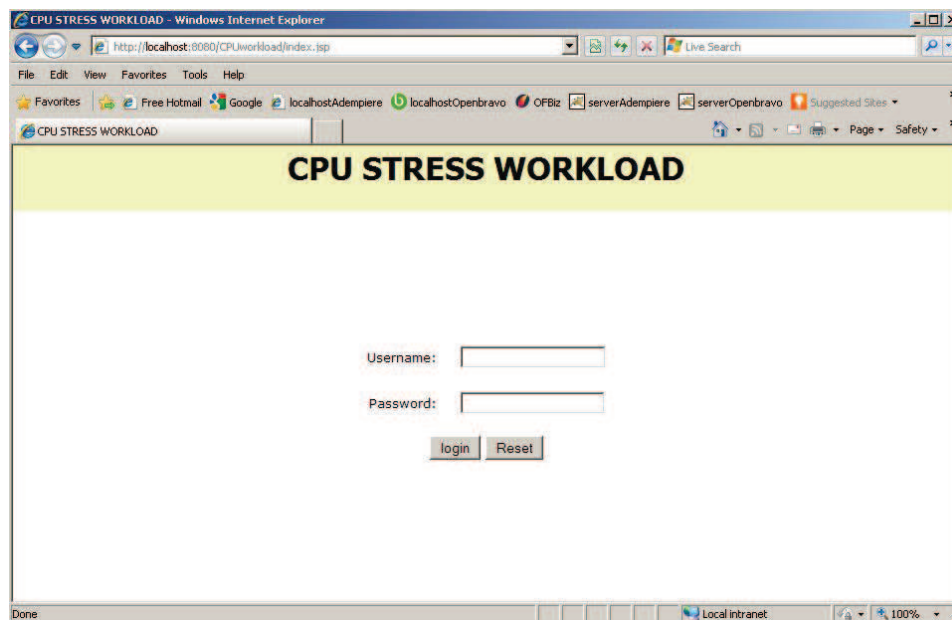


Figura 2.22: Pagina iniziale di autenticazione

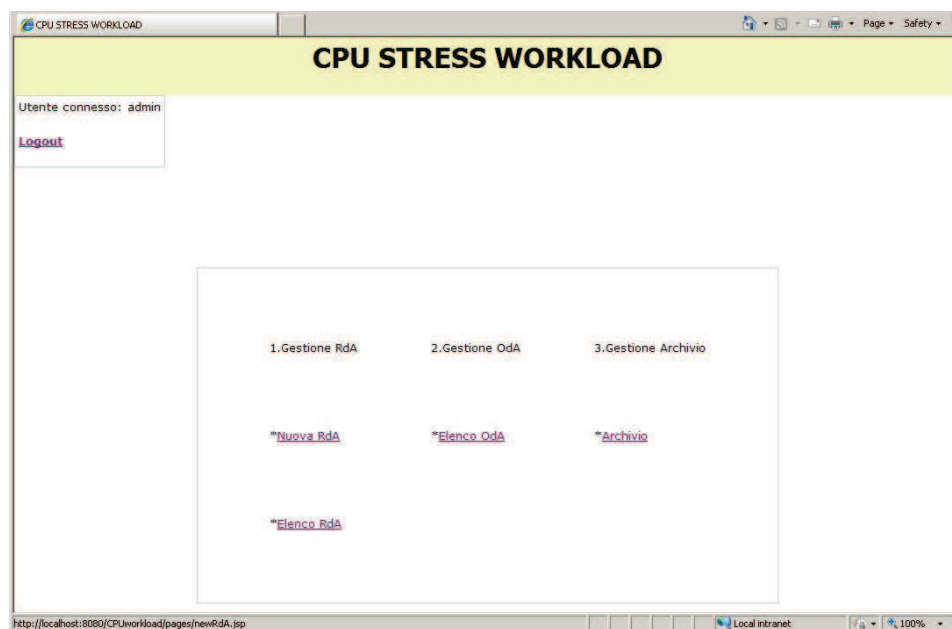


Figura 2.23: Homepage

The screenshot shows a web browser window titled 'Nuova Richiesta di Acquisto' within the 'CPU STRESS WORKLOAD' application. The user is logged in as 'admin'. The form contains the following fields:

- Riferimento Ordine:
- Descrizione:
- Fornitore:
- Data Consegna:
- Indirizzo:
- Prodotto: Qta: Prezzo Unitario: €

A 'Salva' button is located at the bottom of the form.

Figura 2.24: Creazione nuova RdA

The screenshot shows the 'Elenco Richieste di Acquisto' table in the 'CPU STRESS WORKLOAD' application. The table has the following columns: Riferimento RdA, Descrizione, Data Consegna, Indirizzo, Fornitore, Prodotto, Quantità, Importo Totale (€). A single record is visible with the following data:

Riferimento RdA	Descrizione	Data Consegna	Indirizzo	Fornitore	Prodotto	Quantità	Importo Totale (€)
RfOrd1	questa è la descrizione della RfA corrente	2012-05-20	indirizzo di spedizione	fornitore1	Prodotto2	300	27000

Buttons for 'Crea Ordine' and 'Elimina' are visible at the end of the table row. A 'Message from webpage' dialog box is open, asking 'Verrà eliminata una Richiesta di Acquisto. Continuare?' with 'OK' and 'Cancel' buttons.

Figura 2.25: L'RdA creata viene inserita nell'elenco RdA, da cui è possibile creare l'ordine oppure cancellare il record corrispondente.

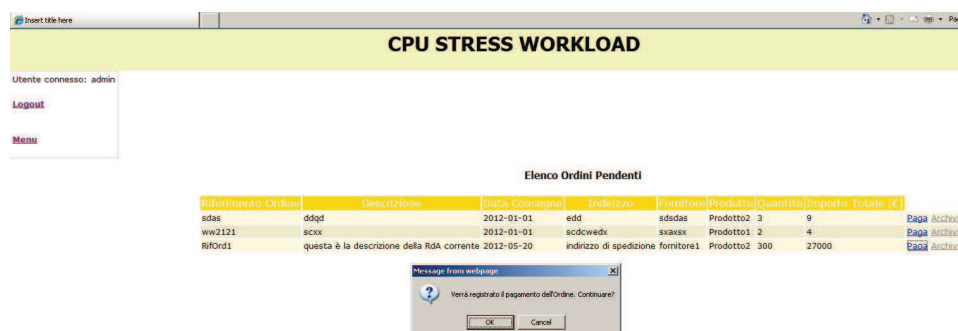


Figura 2.26: L'ordine creato viene inserito nell'elenco Oda pendenti da cui è possibile effettuare il pagamento e, solo successivamente, l'archiviazione. Non è dunque possibile archiviare un ordine prima che venga estinto il pagamento

Elenco Ordini Pendenti

Riferimento Ordine	Descrizione	Data Consegna	Indirizzo	Fornitore	Prodotto	Quantità	Importo Totale (€)	
sdas	ddqd	2012-01-01	edd	sdsdas	Prodotto2	3	9	Paga Archivia
ww2121	scxx	2012-01-01	sdcwedx	svaxsx	Prodotto1	2	4	Paga Archivia
RifOrd1	questa è la descrizione della RdA corrente	2012-05-20	indirizzo di spedizione	fornitore1	Prodotto2	300	27000	Paga Archivia

Figura 2.27: Ordine pagato e pronto per essere archiviato

Archivio Ordini

Riferimento Ordine	Data Consegna Effettuata	
ddqd	2013-07-27	Dettagli
sssss	1906-08-05	Dettagli
RifOrd1	2012-05-20	Dettagli

Figura 2.28: Archiviazione dell'ordine

2.6 Classificazione energetica

Concludiamo il capitolo presentando una metodologia di classificazione energetica del software a partire dal consumo di risorse. Classificare il software in base all'assorbimento di risorse, che si trasformerá poi in assorbimento di energia secondo le caratteristiche di efficienza specifiche dell'hardware, può rappresentare una strada da seguire per avere una indicazione dell'efficienza energetica del software. Classificare energeticamente un software consiste nel cercare di capire cosa il software mi permette di fare e quante risorse esso mi consuma per raggiungere il suo scopo. Successivamente, cercare di trovare una metodologia di paragone con altre soluzioni software che mi permettano di ottenere lo stesso risultato e analizzare i diversi comportamenti energetici o di consumo risorse. [5] Si pensi alla classificazione energetica degli elettrodomestici in cui si definiscono livelli di efficienza basati sui consumi dei vari modelli. Possiamo pensare alla classificazione energetica come all'analisi, all'interno di una determinata classe di "prodotti" (elettrodomestici, abitazioni, pc, hardware), di risultati di determinati test energetici standard. La metodologia qui proposta risulta un'estensione degli studi condotti in [5], in cui si é analizzata e valutata l'efficienza interna di un software enterprise. Si definisce classificazione energetica interna l'individuazione di classi di equivalenza all'interno di una stessa soluzione software in cui sono raggruppate le transazioni di business secondo dei criteri, prestabiliti in base allo scopo della classificazione stessa: funzionale, pesantezza computazionale, prestazionale ecc.. In questo lavoro si é definita una metodologia di classificazione energetica esterna, valutando l'efficienza energetica di una soluzione enterprise rispetto ad altre soluzioni ad essa paragonabili. Per classe di equivalenza, quindi, intendiamo una categoria di applicazioni confrontabili secondo alcuni parametri. Come già descritto all'inizio di questo lavoro, la classe di equivalenza qui considerata é quella dei software enterprise open-source, scritti in linguaggio Java e funzionalmente comparabili. La metodologia di classificazione prevede i seguenti passi:

- Suddivisione delle applicazioni in classi predefinite con caratteristiche omogenee (ad esempio classe dei software enterprise open source);
- Misurazione del consumo di risorse assorbite dalle applicazioni in scenari d'uso comuni;
- Individuazione all'interno di ciascuna classe del valore medio μ e della deviazione standard σ della risorsa misurata;
- A partire da questi dati, individuare degli intervalli di consumo determinati dai seguenti valori soglia:

$$- \mu - \sigma$$

$$- \mu + \sigma$$

- Classificare le applicazioni della classe secondo la seguente scala:
 - Green, se $\gamma < (\mu - \sigma)$
 - Yellow, se $\gamma \in [\mu - \sigma; \mu + \sigma]$
 - Red, se $\gamma > (\mu + \sigma)$

Si noti che il passo 2 sfrutta la metodologia di stima del consumo energetico definita in questo lavoro. La misura del consumo di CPU di un'applicazione é data infatti dalla somma dei consumi delle singole transazioni considerate. Nel capitolo seguente si mostra, a titolo d'esempio, la classificazione di questa metodologia per i software ERP analizzati in questo lavoro.

Capitolo 3

Misure ed Analisi dei Risultati

In questo capitolo si mostrano i risultati delle misure di consumo energetico ed assorbimento di risorse che ci permetteranno di validare la metodologia introdotta.

3.1 Misura dell'idle

La prima misura effettuata riguarda l'assorbimento di potenza del server in stato di idle. Al fine di ottenere una stima attendibile, sono stati effettuati tre esperimenti di misura della durata ciascuno di dieci minuti. Dopodichè si è proceduto a ricavare la stima finale tramite media campionaria dei valori misurati.

Dai dati in tabella si ottiene un valore di potenza medio di 296,09 Watt.

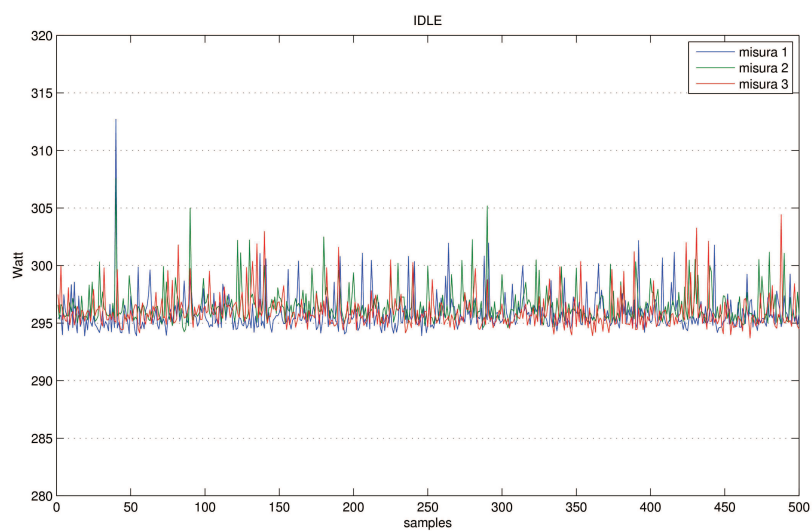


Figura 3.1: Misura dell'idle

	Durata [s]	Potenza media misurata [Watt]
Misura 1	600	295,82
Misura 2	600	296,48
Misura 3	600	295,97

Figura 3.2: Risultati delle misure in idle

3.2 Stima metrica beta

Come descritto nel capitolo precedente, il lavoro di identificazione della caratteristica del processore e stima della metrica beta è avvenuto in tre fasi, migliorando di volta in volta il risultato ottenuto:

- Stima tramite benchmark di stress CPU OCCT
- Stima tramite benchmark di stress Prime95
- Stima tramite web application

In questa sezione si presentano i risultati ottenuti.

3.2.1 OCCT Test

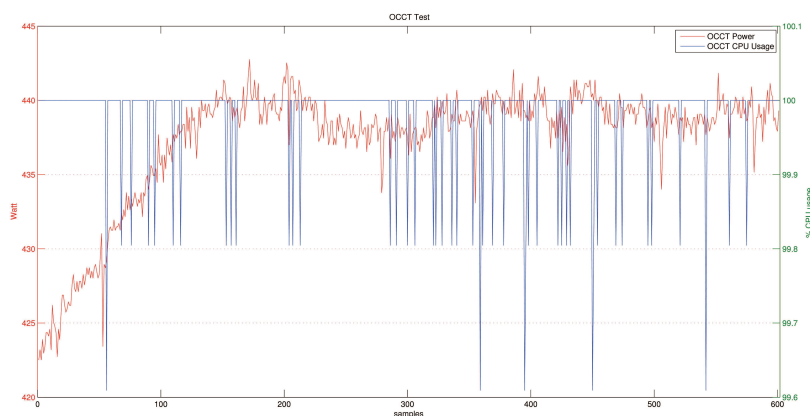


Figura 3.3: OCCT Test

Si noti la presenza di un transitorio iniziale durante il quale l'assorbimento di potenza risulta inferiore ai valori di regime pur essendo l'utilizzo del processore quasi sempre al cento per cento.

Il test OCCT fornisce tuttavia un solo punto sperimentale della caratteristica del processore. Tale punto coincide con l'assorbimento di potenza in condizioni di utilizzo massimo. Questa informazione, combinata con la potenza in idle ottenuta dalla misura precedente, consente l'identificazione della caratteristica sperimentale di figura 3.5.

La metrica β è data dal coefficiente angolare della caratteristica lineare raffigurata ed è pari a 1,413.

	idle	OCCT running
Potenza assorbita [Watt]	296,09	437,37
CPU Utilizzo [%]	0	100

Figura 3.4: Punti stimati con OCCT

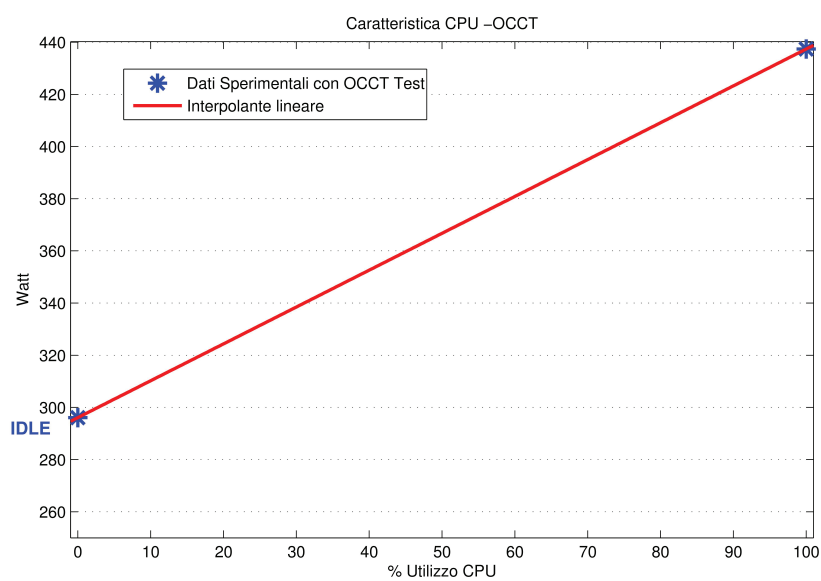


Figura 3.5: Caratteristica ottenuta con OCCT

3.2.2 Prime95 Test

Come detto, il benchmark Prime95 offre la possibilità di eseguire da 1 ad 8 thread in parallelo, essendo 8 il numero di core della macchina di test. Si riportano i grafici delle misure di potenza alle diverse percentuali di utilizzo, a partire dalle quali si sono individuati gli otto punti della caratteristica.

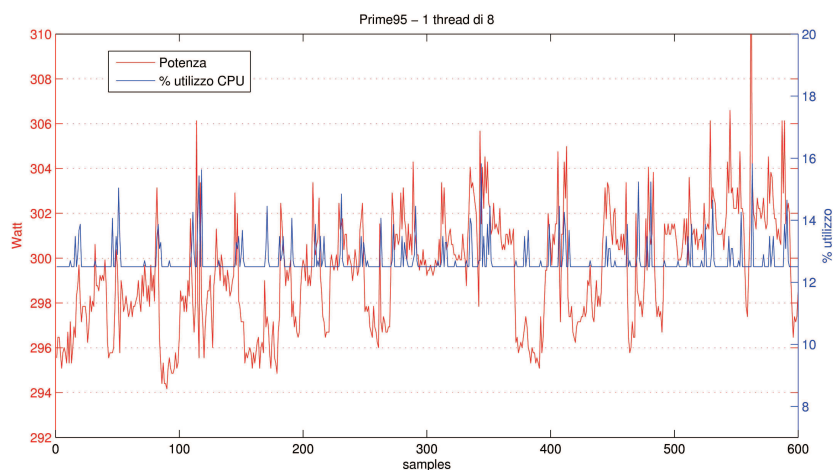


Figura 3.6: Prime95 - 1 core

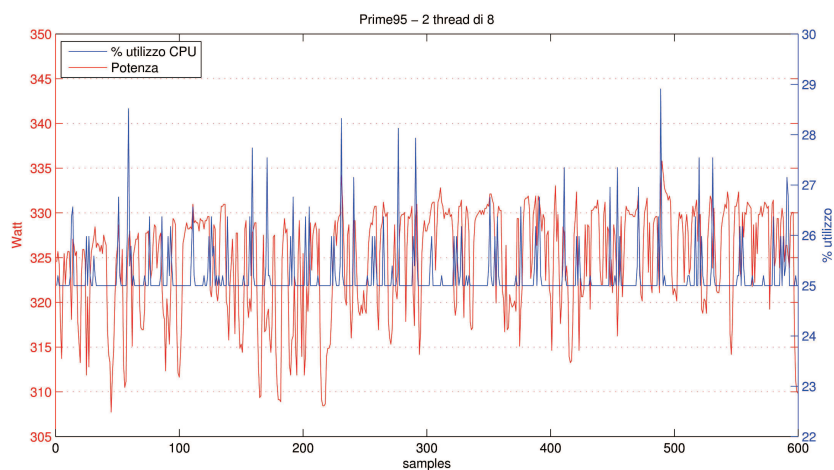


Figura 3.7: Prime95 - 2 core

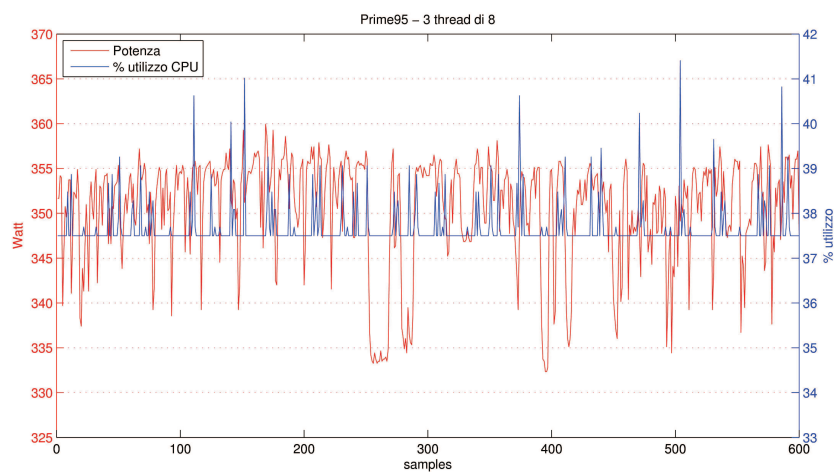


Figura 3.8: Prime95 - 3 core

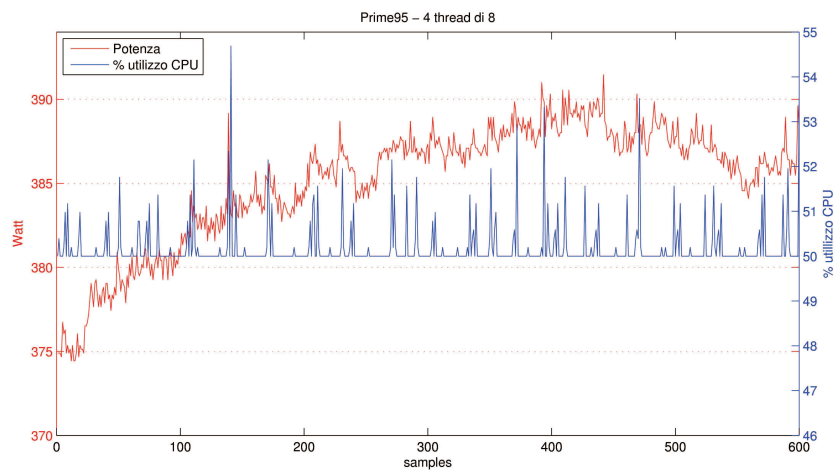


Figura 3.9: Prime95 - 4 core

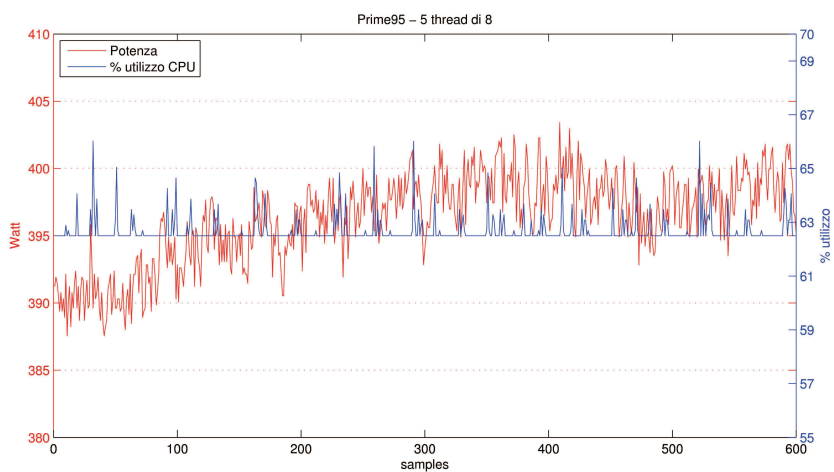


Figura 3.10: Prime95 - 5 core

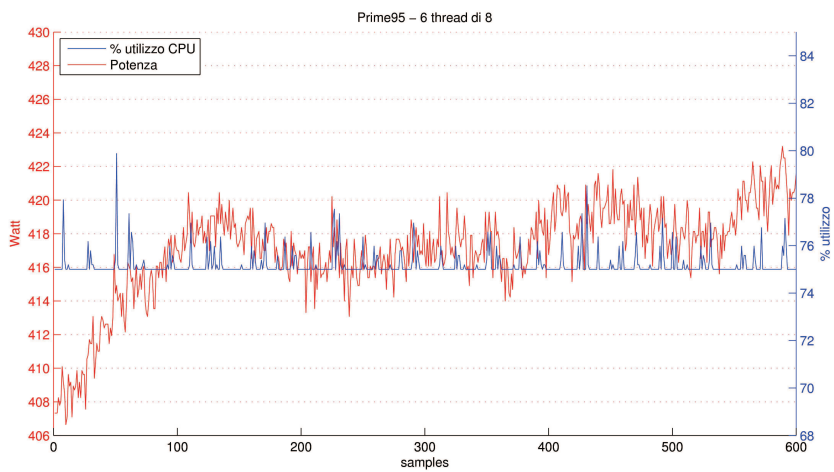


Figura 3.11: Prime95 - 6 core

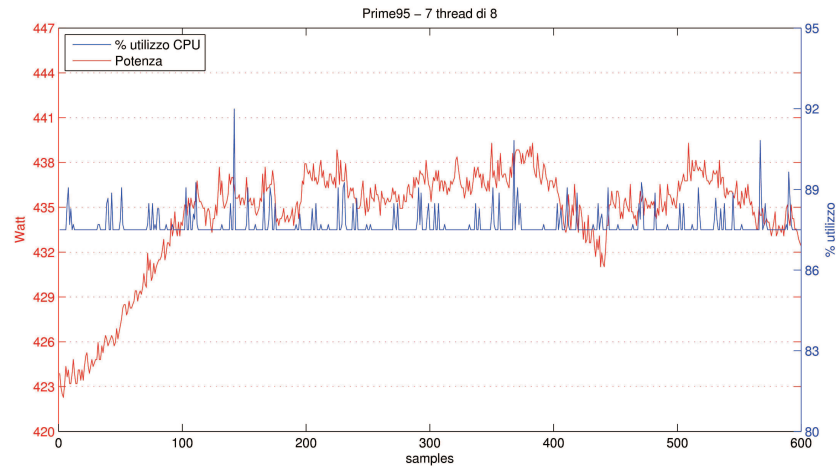


Figura 3.12: Prime95 - 7 core

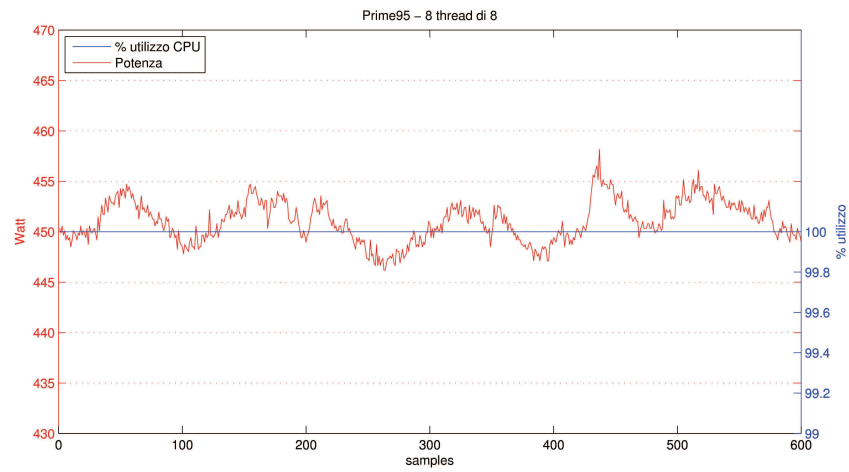


Figura 3.13: Prime95 - 8 core

Le misure effettuate hanno portato all'individuazione dei seguenti otto punti sperimentali da cui si è ottenuta, tramite regressione lineare, la nuova curva caratteristica di figura 3.15.

% Utilizzo	Potenza
12,69	299,37
25,20	324,68
37,67	350,56
50,18	385,14
62,68	396,32
75,19	417,07
87,67	434,46
100,00	451,03

Figura 3.14: Punti stimati con Prime95

La caratteristica del processore, ottenuta tramite regressione lineare, viene qui riportata.

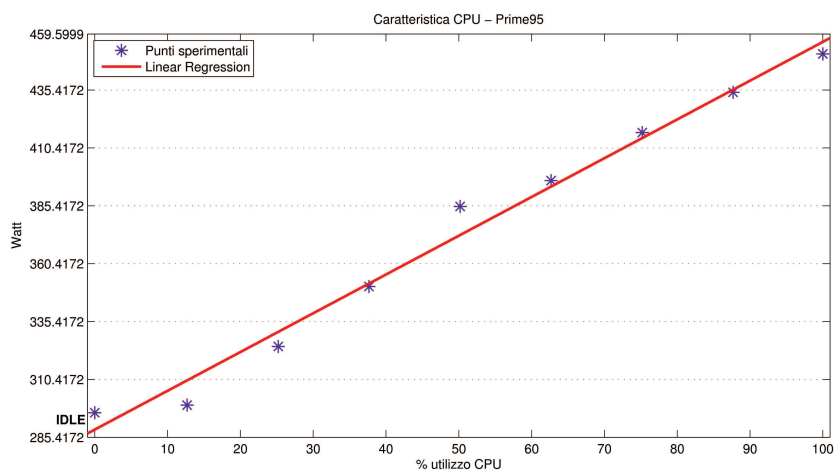


Figura 3.15: Caratteristica ottenuta con Prime95

La metrica beta risultante è pari a 1.674. Si osserva dunque un valore della metrica superiore a quello ottenuto con il benchmark OCCT. Ciò significa che a parità di utilizzo della CPU, il consumo energetico è maggiore durante i test con Prime95. Questo risultato tuttavia non ci sorprende in quanto, come descritto nel capitolo precedente, il benchmark Prime95 esegue stress tests più efficaci rispetto ad OCCT in termini di computazioni eseguite dalla FPU. Sperimentalmente si è dunque verificato che carichi onerosi dal punto di vista computazionale sono più esigenti dal punto di vista energetico rispetto a carichi meno onerosi.

3.2.3 Web Application Test

Analogamente a quanto fatto nei paragrafi precedenti, si presentano ora i risultati delle misure effettuate per la stima della metrica beta attraverso la web application che è stata presentata nel capitolo precedente. Al fine di stressare il processore si è implementato uno script LoadRunner che effettua le seguenti azioni:

1. Creazione di 3 richieste di acquisto
2. Creazione di 2 ordini di acquisto
3. Pagamento e archiviazione degli ordini di acquisto creati

Successivamente sono state effettuate diverse prove di carico con lo script realizzato, utilizzando il Controller di LoadRunner. L'obiettivo in questa fase è stato quello di individuare quali carichi generassero utilizzi costanti per un tempo sufficientemente lungo. In figura 3.16 riporta il grafico di una simulazione eseguita.

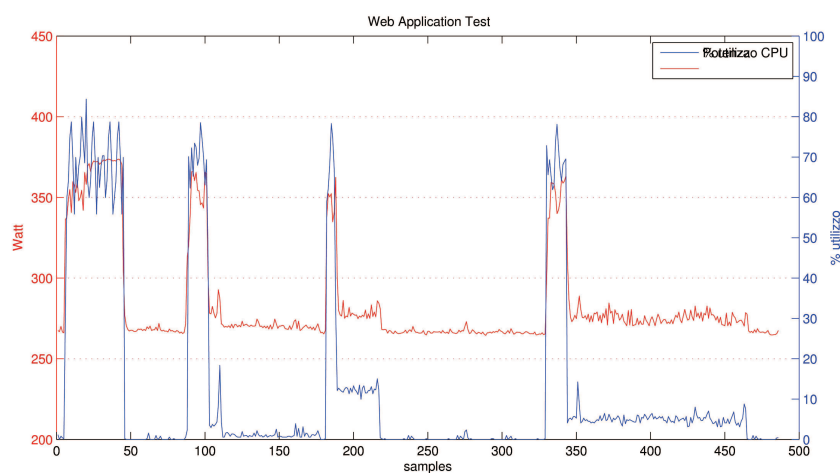


Figura 3.16: Potenza ed utilizzo CPU tramite la Web Application

Da questo grafico 3.16 sono stati individuati tre punti candidati ad appartenere alla caratteristica.

% Utilizzo	Potenza
4,92	275,19
12,29	290,95
68,55	361,37

Figura 3.17: Web Application - punti candidati

La caratteristica 3.18 comporta una β pari a 1,332.

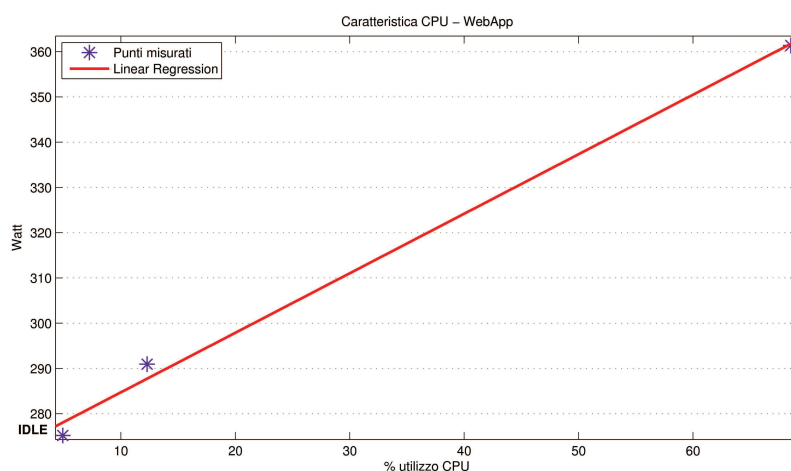


Figura 3.18: Caratteristica identificata tramite la Web Application

3.3 Validazione Modello energetico

Si presentano ora i risultati di misurazione energetica delle transazioni specificate nel nostro benchmark. Tali risultati ci consentiranno di validare la metodologia di misura indiretta del consumo energetico per sistemi ERP. Le misure sono state condotte nelle seguenti condizioni iniziale:

- Tensione di alimentazione 230 Volt
- Potenza in idle pari a 296,06 Watt
- Metrica β pari a 1,332

Per ogni misura, si riportano i grafici di utilizzo processore (da cui si è calcolata la metrica γ) e di consumo di potenza (da cui si è calcolata l'energia realmente consumata). Si riportano infine in forma tabellare i seguenti valori:

- Durata della misura in secondi
- Numero di client concorrenti eseguiti
- Energia misurata totale
- Energia misurata differenziale
- γ : metrica di consumo risorsa CPU
- $\gamma\beta$: energia differenziale stimata tramite la metodologia di misura indiretta

- Scarto: errore in percentuale fra energia differenziale stimata ed energia differenziale misurata

L'energia differenziale misurata è ricavata a partire dall'energia totale e sottraendo l'energia consumata in idle secondo la relazione di base che lega potenza (in idle) ed energia:

$$E_{idle} = P_{idle} * t$$

dove t indica la durata della misura.

3.3.1 Test1: Inserimento nuovo Business Partner

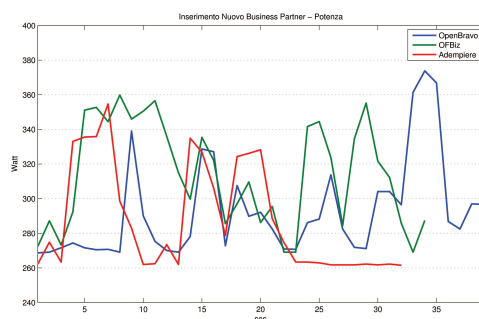


Figura 3.19: Potenza - Nuovo Business Partner

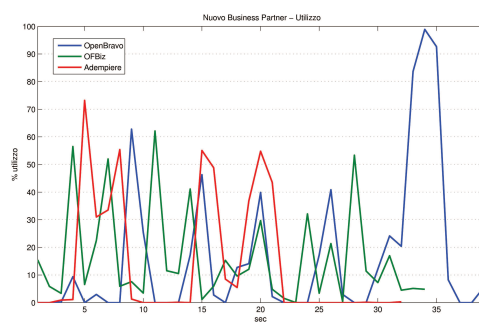


Figura 3.20: Utilizzo - Nuovo Business Partner

	Durata [s]	# client	Energia misurata [Joule]	Energia diff misurata [Joule]	Metrica γ [%*s]	Energia diff stimata [$\gamma \beta$]	Errore [%]
Openbravo	38	10	11129,47	905,19	640,01	853,27	5,74
Adempiere	31	10	8949,65	608,78	450,00	599,94	1,45
OFBiz	34	10	9831,28	683,24	525,88	701,11	2,61

Figura 3.21: Risultati - Nuovo Business Partner

3.3.2 Test2: Inserimento nuovo Prodotto

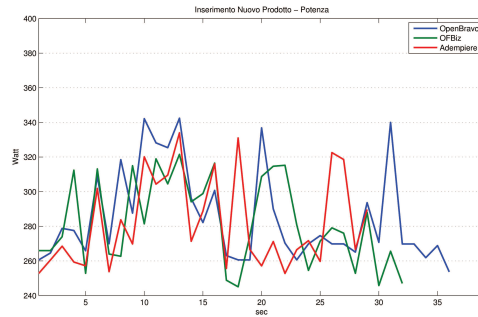


Figura 3.22: Potenza - Nuovo Prodotto

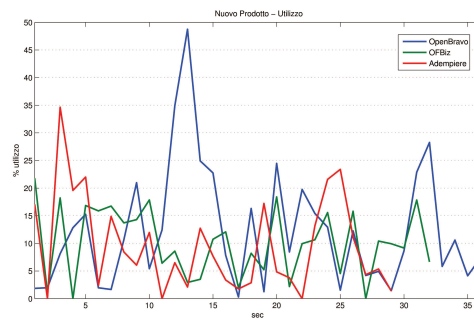


Figura 3.23: Utilizzo - Nuovo Prodotto

	Durata [s]	# client	Energia misurata [Joule]	Energia diff misurata [Joule]	Metrica γ [%*s]	Energia diff stimata [J]	Errore [%]
Openbravo	35	10	10011,34	594,24	438,73	584,92	1,57
Adempiere	28	10	7909,54	375,86	271,10	361,44	3,84
OFBiz	31	10	8773,17	432,31	313,06	417,38	3,45

Figura 3.24: Risultati - Nuovo Prodotto

3.3.3 Test3: Inserimento nuovo Ordine di Acquisto

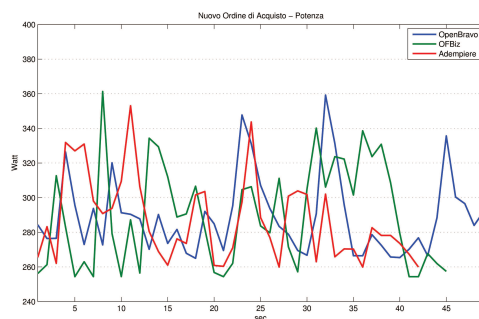


Figura 3.25: Potenza - Nuovo Ordine di Acquisto

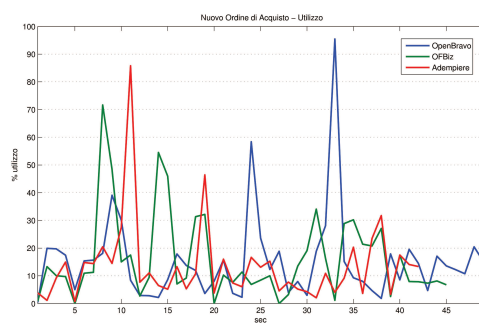


Figura 3.26: Utilizzo - Nuovo Ordine di Acquisto

	Durata [s]	# client	Energia misurata [Joule]	Energia diff misurata [Joule]	Metrica γ [%*s]	Energia diff stimata [γ β]	Errore [%]
Openbravo	48	10	13870,58	955,70	738,99	985,22	3,09
Adempiere	41	10	11790,89	759,43	558,65	744,80	1,93
OFBiz	44	10	12772,90	934,26	725,74	967,56	3,56

Figura 3.27: Risultati - Nuovo Ordine di Acquisto

3.3.4 Test4: Inserimento nuovo Ordine di Vendita

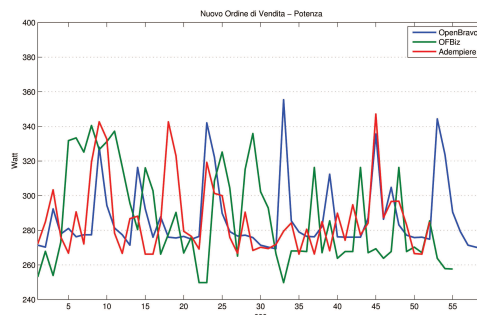


Figura 3.28: Potenza - Nuovo Ordine di Vendita

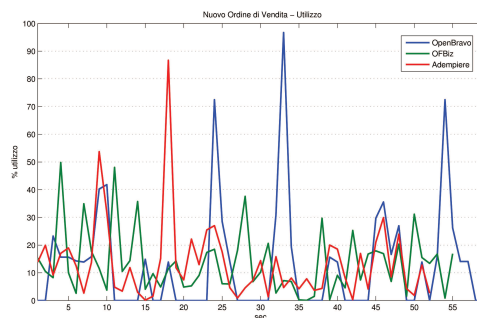


Figura 3.29: Utilizzo - Nuovo Ordine di Vendita

	Durata [s]	# client	Energia misurata [Joule]	Energia diff misurata [Joule]	Metrica γ [%*s]	Energia diff stimata [J β]	Errore [%]
Openbravo	58	10	16651,89	1046,68	766,80	1022,30	2,33
Adempiere	51	10	14621,43	899,61	691,80	922,30	2,52
OFBiz	54	10	15513,78	984,79	732,22	976,20	0,87

Figura 3.30: Risultati - Nuovo Ordine di Vendita

3.3.5 Analisi dei Risultati

Analizzando i nostri risultati si osserva come il modello utilizzato costituisca un buon stimatore del consumo energetico a partire dall'utilizzo del processore. Gli errori registrati infatti rientrano in un range limitato che varia in valore assoluto tra lo 0,87% e il 5,74%. Tali errori, seppur accettabili, sono imputabili principalmente a tre fattori.

In primo luogo le misure di assorbimento di potenza sono relative all'intero server di test e non al solo processore. Questo elemento, sebbene

rappresenti un limite al lavoro di validazione, non ha in ogni caso impedito la verifica della bontà del modello di stima del consumo energetico per i sistemi ERP considerati. Gli errori infatti risultano, come detto, sufficientemente contenuti e confermano la tesi sostenuta in [5] secondo cui circa il 97% dell'energia stimata deriva dall'utilizzo del processore. Il processore è responsabile della quasi totalità dell'energia aggiuntiva che noi andiamo a stimare. L'energia degli altri componenti (quale attività del disco, utilizzo rete, memoria ecc..) è contenuta nella componente di energia di idle, ed influisce minimamente sulla componente energetica aggiuntiva. [5]

Un secondo fattore che ha certamente influenzato i risultati è il parametro beta utilizzato. Se da un lato l'indagine condotta sulla dipendenza del beta dalla tipologia di carico ha consentito l'individuazione (tramite la web application) di un valore certamente attendibile, è anche vero che la caratteristica è stata stimata a partire da soli tre punti. Ciò rappresenta un'approssimazione che certamente impatta sui risultati ottenuti. Gli strumenti utilizzati per la misura di corrente e CPU Time costituiscono un'altra possibile causa degli errori ottenuti. Sia i dati di utilizzo CPU che quelli di potenza sono infatti ottenuti campionando i segnali alla frequenza massima di 1Hz, il che risulta abbastanza limitante per le frequenze di lavoro delle attuali architetture hardware.

Un'ultima considerazione sempre deducibile dai risultati riguarda la durata dei test eseguiti. Dai dati riportati nelle tabelle si nota infatti l'elaborazione con Adempiere sia più veloce in tutti gli scenari di esecuzione considerati. In generale ci si aspetta dunque un risparmio energetico utilizzando questa soluzione come sistema enterprise. A proposito della determinazione dell'efficienza energetica di una soluzione applicativa, si propone nel seguente paragrafo un esempio di classificazione energetica secondo la metodologia presentata nel capitolo precedente.

3.4 Esempio di classificazione energetica

Partendo dai risultati presentati nel paragrafo precedente, poniamo ora l'attenzione sui valori ottenuti per la metrica γ nei diversi scenari simulati.

	Nuovo Business Partner	Nuovo Prodotto	Nuovo Ordine di Acquisto	Nuovo Ordine di Vendita
Openbravo	640,01	438,73	738,99	766,80
Adempiere	450,00	271,10	558,65	691,80
OFBiz	525,88	313,06	725,74	732,22

Figura 3.31: Tabella riassuntiva metriche gamma

La tabella 3.32 mostra il valore medio e la deviazione standard che caratterizzano la classe a cui appartengono i tre sistemi analizzati.

$\mu[\gamma]$	$\sigma[\gamma]$	Soglie $[\gamma]$
571,08	172,32	[398,76 ; 743,4]

Figura 3.32: Parametri statistici derivati

I valori ottenuti portano alla seguente classificazione per ciascuna transazione:

	Nuovo Business Partner	Nuovo Prodotto	Nuovo Ordine di Acquisto	Nuovo Ordine di Vendita
Openbravo	●	●	●	●
Adempiere	●	●	●	●
OFBiz	●	●	●	●

Figura 3.33: Esempio di classificazione energetica per i tre sistemi ERP

Possiamo dunque concludere che Openbravo risulta un sistema complessivamente meno efficiente dal punto di vista energetico rispetto ad Adempiere ed OFBiz che hanno invece un comportamento simile.

Capitolo 4

Conclusioni e Sviluppi Futuri

Il lavoro condotto rientra in un ampio progetto di ricerca che mira a studiare nuove metodologie nell'ambito del Green Software, per l'analisi e valutazione indiretta dell'efficienza energetica del software, ovvero a partire dal consumo di risorse dei sistemi.

La prima parte del lavoro é consistito nello studio di una recente metodologia basata sull'analisi dell'utilizzo del processore durante l'esecuzione di software enterprise e sulla valutazione delle performance energetiche di una macchina server.

In questa prima parte si é evidenziato il grosso limite del modello precedente, il quale considera il parametro che modella le performance del server, chiamato metrica beta, indipendente dalla tipologia di carico a cui é sottoposto il processore.

Il contributo che si é apportato riguarda la stima di tale metrica, a partire da un carico di tipo transazionale e quindi il piú simile possibile ai sistemi ERP oggetto di questo lavoro. Il carico transazionale é rappresentato da una serie di operazioni di lettura e scrittura su database, eseguite da una applicazione java web-based che é stata implementata e che simula l'operativitá di un processo di eProcurement.

I risultati relativi alla validazione del modello hanno infatti evidenziato come, utilizzando la nostra nuova stima della metrica beta, l'errore percentuale si riduca notevolmente e si assesti in un range tra lo 0,87% e il 5,74%. Diversamente, la stima della metrica beta, a partire da carichi funzionalmente differenti da quello transazionale, falsa il lavoro di validazione introducendo degli errori sensibilmente maggiori.

La validazione del modello é stata condotta a partire da un benchmark specifico per soluzioni enterprise open source, scritte con linguaggio java e supporto per database PostgreSQL.

L'obiettivo di questa seconda parte di lavoro é stato l'identificazione di un set di operativitá il piú possibile indipendenti dalla specifica soluzione. Que-

sta fase si é concretizzata nella definizione di quattro transazioni di business a partire da tre soluzioni di sistemi ERP: Openbravo, Adempiere e Apache OFBiz. La specifica del benchmark si é completata poi con l'implementazione di scenari di esempio di tali transazioni, realizzati con il supporto di HP LoadRunner, con cui sono state effettuate le misure di test. L'utilizzo di una pinza amperometrica e di un multimetro digitale ha consentito infine l'acquisizione dei dati di corrente tramite cui é stato condotto il lavoro di validazione.

Evidenze sperimentali hanno mostrato come un'ulteriore indagine sia necessaria per quanto riguarda l'analisi dei consumi energetici del software, studiando per esempio l'impatto di altri fattori quali la dimensione dei dati di input, la dimensione del set informativo su database o il numero di utenti connessi. Per esempio, dal punto di vista del consumo energetico é la stessa cosa modificare la giacenza a magazzino di un prodotto quando ci sono 10 tipologie di prodotti presenti o quando ce ne sono 1.000? Qual é la variazione di assorbimento energetico relativa ad una variazione nel numero di utenti che si connettono all'applicativo? Un ulteriore spunto di riflessione infine riguarda lo studio del reale contributo di altri componenti quali memoria e interfaccia di rete e la definizione di un modello piú accurato che si basi sull'identificazione di una curva di performance del server che si avvicini il piú possibile al suo reale comportamento energetico.

Studi recenti hanno mostrato a questo proposito che un'approssimazione logaritmica determina una riduzione di almeno un punto percentuale sugli errori di validazione del modello.

Come ultimo passo si ritiene interessante approfondire il concetto di classificazione energetica del software, brevemente introdotto in questo lavoro proponendo un semplice criterio di classificazione su base statistica per la classe dei sistemi ERP. Un interessante approfondimento potrebbe essere l'utilizzo e il miglioramento di tale metodologia di classificazione per altre classi di applicazioni, quali applicativi desktop, sistemi CRM.. Una buona metodologia di classificazione puó senz'altro contribuire nelle scelte del manager in fase di software selection in ottica di un sistema sempre piú green e complessivamente piú efficiente.

Appendice A

Il Benchmark TPC-C

La specifica TPC-C è stata introdotta dal Transaction Processing Performance Council nell'Agosto del 1992 con l'obiettivo di sostituire i benchmark TPC-A e TPC-B come soluzione di benchmarking per ambienti OLTP. Include e definisce un set di attività (transazioni) core in un tipico ambiente di business nel contesto della gestione degli ordini di un fornitore all'ingrosso. Tuttavia la specifica non è definita per un solo segmento di business, ma si rivolge a qualsiasi ambiente enterprise che compra, vende, produce e distribuisce prodotti o servizi. In particolare le operazioni considerate riguardano:

- consegna degli ordini in entrata
- registrazione dei pagamenti
- controllo stato degli ordini
- monitoraggio livello stock a magazzino

Il benchmark TPC-C è caratterizzato da:

- Simulazione dell'esecuzione simultanea di transazioni multiple
- Differenti tipi di transazioni a differenti livelli di complessità
- Modalità di esecuzione transazioni on-line e asincrona tramite sistema a coda
- Sessioni on-line multiple
- Significativo utilizzo del disco in lettura e scrittura
- Integrità delle transazioni rispetto alle proprietà ACID (atomicità, consistenza, isolation, durabilità)
- Distribuzione dei dati di accesso tramite chiavi primarie e secondarie non uniforme

- Database consistente in molte tabelle di grandi dimensioni, con diversi attributi e relazioni
- Contention sull'accesso e aggiornamento dei dati

La metrica per la misura delle performance definita nella specifica è il “business throughput” ovvero il numero di ordini processati al minuto ed è espressa in transazioni-per-minuto (tpmC). Questo perché le transazioni sono soggette a vincoli sul tempo di risposta. Altre due metriche definite nella specifica sono la “Price/Performance metric” e la “Availability Date metric”.

Al fine di essere compliant con l'estensione TPC-Energy standard, è definita una metrica addizionale espressa in watts-per-tpmC. La specifica TPC-C prevede un database costituito da nove tabelle con cardinalità fissata. Il modello dei dati è definito a partire da un magazzino (W) a cui fanno riferimento 10 punti di vendita e 3000 clienti associati ad ogni punto vendita. Ogni magazzino mantiene uno stock per ognuno dei 100000 prodotti offerti. In figura si mostra il modello dei dati della specifica.

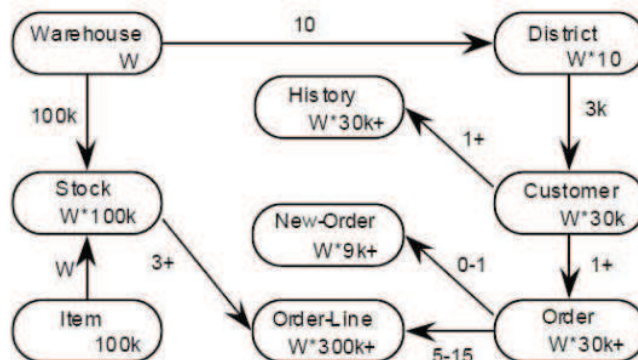


Figura A.1: Modello dei dati della specifica TPC-C

I clienti usano il sistema per fare il placement di un ordine o per richiedere lo stato di un ordine esistente. Gli ordini sono composti da una media di 10 order lines. Il sistema è inoltre utilizzato per processare il pagamento e la consegna degli ordini e per esaminare i livelli di stock a magazzino. L'1% di tutti gli ordini sono relativi ad item non presenti nel magazzino locale e devono essere processati tramite un altro magazzino. Il “core” della specifica riguarda l'implementazione delle transazioni. Esistono cinque differenti transazioni, di seguito analizzate.

New Order Transaction consiste nell'inserimento di un ordine attraverso una singola transazioni su database. È progettata per rappresentare un carico medio in lettura/scrittura con vincoli sul response time. La

sua specifica riflette la tipica attività on-line basata su database di un ambiente di produzione.

Payment Transaction aggiorna il bilancio del cliente e riporta il pagamento sulle statistiche di vendita. Rappresenta un carico medio-basso in lettura/scrittura con elevati frequenze di esecuzione e vincoli sul response time. Include l'accesso alla tabella Customer basato su chiave non primaria.

Order-Status Transaction è una transazione di business per l'interrogazione sullo stato dell'ultimo ordine effettuato dal cliente. Rappresenta un carico medio in sola lettura con bassa frequenza di esecuzione e vincoli sul response time. Come la Payment Transaction, include l'accesso alla tabella Customer basato su chiave non primaria.

Delivery Transaction consiste nel processamento di 10 nuovi ordini in modalità batch. Ogni ordine è processato completamente in una singola transazione su database. Il numero di ordini processati per singola transazione su database è dipendente dall'implementazione del benchmark. La Delivery Transaction ha una frequenza di esecuzione bassa e requisiti sul response time meno stringenti. La modalità di esecuzione è asincrona tramite meccanismo di queueing con segnalazione finale di completamento della transazione.

Stock-Level Transaction determina tutti i prodotti con livello di stock sotto una specifica soglia. Rappresenta un carico elevato in sola lettura, con frequenza di esecuzione bassa e vincoli rilassati sia sul response time che sulla consistenza delle informazioni.

La frequenza di occorrenza di ciascuna transazione è riportata in tabella.

Transaction	Frequency
New-Order	44,0%
Payment	44,0%
Order-Status	4,0%
Delivery	4,0%
Stock-Level	4,0%

Figura A.2: Distribuzione frequenza della transazioni

Le percentuali sono fissate in modo da garantire di avere circa una "Payment Transaction" per ogni "New Order Transaction" e circa una "Order-Status Transaction", una "Delivery Transaction" e una "Stock-Level Transaction" ogni 10 "New Order Transaction".

Bibliografia

- [1] European Commission - Information Society and Media DG, Final Report - September 2008
Impact of Information and Communication Technologies on Energy Efficiency

- [2] Jens Vykoukal, Roman Beck, Martin Wolf,
Impact of pressure for environmental sustainability on grid assimilation - empirical results from the financial service industry

- [3] Eva Kern, Markus Dick, Timo Johann, Stefan Naumann
Green Software and Green IT: An End User Perspective

- [4] W.Fornaciari , P.Gubian, D.Sciuto, C.Silvano
Power estimation of embedded systems: A hardware/software codesign approach

- [5] F.Di Domizio
Definizione e validazione Definizione e validazione empirica di una empirica di una metodologia per la stima dei consumi, dell'efficienza energetica e per il benchmarking energetico del software applicativo

- [6] International Telecommunication Union, November 2010
Using ICTs to tackle climate change

- [7] The Climate Group report,
Delivering Low Carbon Growth - A Guide to China's 12th Five Year Plan

-
- [8] Eugenio Capra, Giovanni Agosta - Mondo Digitale, Marzo 2011
Green Software, anche le applicazioni consumano energia
- [9] Eugenio Capra, Chiara Francalanci, Giulia Formenti, Stefano Gallazzi
The impact of MIS Software on IT Energy Consumption
- [10] Brown E.G., Lee C. - Forrester Research Report, November 2007
Topic Overview: Green IT
- [11] Filippo Bonoli - Tesi di Laurea
Una metodologia per ridurre il consumo energetico del software, ed il suo impatto sul Total Costo di Ownership
- [12] Wikipedia
Scienza Olistica
- [13] F. Renzi - Presentazione IBM, Conferenza Green ICT, Novembre 2007
Milano
Evolution scenarios for data centers technologies and architectures and impacts on energy consumptions
- [14] Restorik T., Plan Report, 2007
An Inefficient Truth. Global Action
- [15] Giovanna Sissa, Mondo digitale, Settembre 2009
Green Software
- [16] M.Poess, R.Othayoth, K.Vaid, J.M.Stephens, K.Huppler, E.Haines
Energy Benchmarks: A Detailed Analysis
- [17] Avneesh Saxena, Daphne Chung, April 2009
Optimizing the Datacenter for Cost and Efficiency
- [18] Specifica Benchmark SPC-1/E
<http://www.storageperformance.org/specs>

-
- [19] Erik Young, Paul Cao, Mike Nikolaiev
First TPC-Energy Benchmark: Lessons Learned in Practice
- [20] Andrew Fanara, Evan Haines, Arthur Howard
The State of Energy and Performance Benchmarking for Enterprise Servers
- [21] San Murugesan,
Harnessing Green IT: Principles and Practices
- [22] Klaus-Dieter Lange, IEEExplore Digital Library
Identifying Shades of Green: The SPECpower Benchmarks
- [23] Eugenio Capra, Chiara Francalanci
Green IT: Sfide e Opportunità
- [24] The Climate Group report
SMART 2020: Enabling the low carbon economy in the information age
- [25] Albert Huang
A model for environmentally sustainable information systems development
- [26] Telecommunications Industry Association
TIA's 2011 ICT Market Review & Forecast
- [27] Kumar R. Gartner report, Gennaio 2007
Important Power, Cooling and Green IT Concerns
- [28] Stefan Neumann
Sustainability Informatics: a new Subfield of Applied Informatics?
- [29] Stefano Gallazzi, Giulia Formenti
A methodology to evaluate empirically software energy consumption and its impact on Total Cost of Ownership

-
- [30] Meikel Poess, Raghunath Othayoth Nambiar
Energy Cost, The Key Challenge of Today's Data Center: A Power Consumption Analysis of TPC-C Results
- [31] Gabriele Galli. Politecnico di Milano, Aprile 2009
Un approccio alla valutazione dei consumi energetici basato sull'analisi statica del codice
- [32] Wikipedia
Sviluppo sostenibile
- [33] Eugenio Capra, Chiara Francalanci, Sandra Slaughter
Measuring Application Software Energy Efficiency
- [34] Jens Vykoukal, Roman Beck, Martin Wolf,
Impact of Pressure for environmental sustainability on grid assimilation - Empirical results from the financial services industry
- [35] Antonio Amalfi, Tomas Aurora Vagnato. Politecnico di Milano, 2009
Una metodologia empirica per analizzare l'impatto della struttura del codice sul consumo energetico indotto dalle applicazioni software
- [36] Eugenio Capra, Francesco Merlo
Green IT: Everything starts from the software