Speed-up nei Sistemi di Elaborazione: Principali Tecniche

Pierfrancesco Foglia

Università di Pisa

foglia@iet.unipi.it
http://garga.iet.unipi.it

Introduzione (I)

- Obiettivo:
  - Descrivere le principali tecniche off-the-shelf per aumentare le prestazioni di un sistema di elaborazione
    - A componenti discreti
    - Con il semplice inserimento e senza riprogettazione
- Metriche:
  - La principale: il **tempo di esecuzione**

$$Texe$$

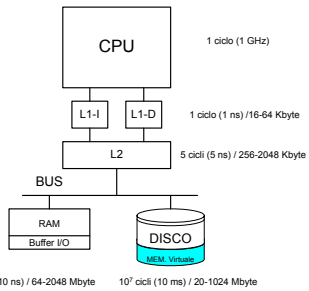
    - Viene misurato/stimato rispetto ad un benchmark
  - Lo **speed-up**:

$$Speedup = \frac{Texe_{base}}{Texe_{opt}} \geq 1$$

    - Valuta l'efficacia di una soluzione

Architettura di riferimento (I)

- Architettura a più livelli di memoria



  - Fonte: Tanenbaum, Modern Operating System, 2nd Edition

Architettura di riferimento (II)

- Caratteristiche:
  - Cache L1 I-D
  - Cache L2 unified
  - Memoria Principale (RAM-DRAM)
  - Memoria Virtuale (DISCO)
  - Buffer della memoria non volatile (RAM)
  - Memoria non volatile o secondaria (DISCO)
- Modello valido per macchine con operativi multiprogrammati
  - UNIX nelle varie versioni (workstation, server)
  - WINDOWS NT/2000/2003 (workstation, server)

Effetti delle gerarchie di memoria (I)

- **Componenti del tempo di esecuzione**

$$Texe = T_{busy} + T_{idle} + T_{other}$$

  - Tbusy: la cpu effettua del lavoro utile
  - Tidle: la cpu è idle per effetto della memoria
  - Tother: la cpu è impegnata per altri task
  - **Una applicazione è può composta da più processi e più applicazioni possono essere in esecuzione**
- **Cache**
  - Il tempo di risoluzione di una miss si accumula in Tidle
    - Con alcune avvertenze (write buffer, OOO, etc.)

Effetti delle gerarchie di memoria (II)

- **Memoria**
  - Un page-fault determina un cambiamento di contesto
    - Tidle aumenta se non ci sono processi Ready
    - Tother aumenta se il processo non appartiene alla applicazione
- Disco
  - Un accesso al disco determina un cambiamento di contesto
    - Tidle aumenta se non ci sono processi Ready
    - Tother aumenta se il processo non appartiene alla applicazione
  - **In generale ciò è vero per tutti i dispositivi di I/O**

Speed-up del modello base (I)

- Cache
  - Aumentandone la dimensione diminuisce il miss-rate
    - Fino ad un certo limite (cold-miss, etc)
  - L1: solitamente è integrata nel chip per motivi di prestazioni
    - Non si può aumentare in modo off-the-shelf
  - L2:
    - nei processori recenti è integrata nel package (Pentium IV)
    - Non si può aumentare in modo off-the-shelf
  - Provare a disabilitare la cache in un PIV

Speed-up del modello base (II)

- Memoria
  - Effetti dell'incremento delle dimensioni:
    - Diminuiscono i page-fault
    - Aumenta la dimensione dei buffer del disco
    - Diminuiscono Tidle e Tother
    - **Lo speed-up può essere significativo perchè si diminuiscono gli accessi al disco**
  - Effetti dell'incremento della velocità:
    - Deve essere compatibile con i timing del bus
    - Diminuisce Tidle (per le miss) ed indirettamente Tother

Speed-up del modello base (III)

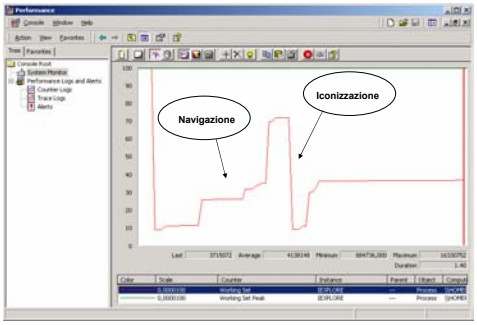
- Fino a che punto aumentare la dimensione della RAM ?
  - Occorre determinare il working set dell'applicazione (TUNING)
  - Definizione di working set: Da MICROSOFT

    "The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed they will then be soft-faulted back into the Working Set before they leave main memory."

  - Il working set dell'applicazione deve essere residente in RAM
  - Opportuni tool permettono di misurare il Working Set
    - Es: Performance Monitor di Windows 2000

Speed-up del modello base (IV)

- Esempio: WS di IE in una sessione di navigazione



Speed-up del modello base (V)

- Dal grafico del WS rilevo quali sono le richieste di memoria dell'applicazione.
- In generale, conviene graficare l'utilizzazione della CPU insieme a disco e RAM
  - Elevata attività del disco con bassa utilizzazione della CPU è sintomo di scarsa RAM disponibile o cattivo tuning del sistema

Speed-up del modello base (VI)

- Per quali applicazioni lo speed-up è significativo?
  - DB
    - La dimensione dell'area condivisa (cache in RAM del DB) è un parametro di configurazione del DB
    - Query OLTP traggono il massimo beneficio
    - Per DSS è inferiore
      - Ranghanatan, ISCA98
  - Server Web statici:
    - Non molto significativo
  - Server Web dinamici (includono DB):
    - Come i DB

Speed-up del modello base (VI)

- Applicazioni Java/Corba e DataCenter
  - Si, per la necessità di avere in esecuzione istanze della JVM o del framework
- Applicazioni grafiche
  - Sicuramente
- Applicazione Scientifiche
  - Dipende dalla dimensione del problema

Speed-up del modello base (VII)

- DISCO
  - Aumento della dimensione?
    - Non è significativo
  - Aumento della velocità:
    - Diminuiscono Tidle e Tother
    - Si ricorre al parallelismo:
      - Sistemi RAID con stripe-set
      - Anche per la fault-tolerance:
        - » RAID con Mirroring o Stripe-set con parità
      - Può essere implementato in hw e sw
        - » Hw più costoso ma più efficiente
        - » Esempio: windows NT/2000
  - Applicazioni che si avvantaggiano:
    - DBMS – DSS e DataCentric (con grosse moli di dati riutilizzate)

Speed-up del modello base (VIII)

- RAID

RAID Level	RAID Type	RAID Description	Advantages & Disadvantages
5+1	Disk striping with parity + mirroring	Six or more volumes, each on a separate drive, are configured identically as a mirrored stripe set with parity error checking.	Provides very high level of fault tolerance but has a lot of overhead.
5	Disk striping with parity	Three or more volumes, each on a separate drive, are configured as a stripe set with parity error checking. In the case of failure, data can be recovered.	Fault tolerance with less overhead than mirroring. Better read performance than disk mirroring.
1	Disk mirroring	Two volumes on two drives are configured identically. Data is written to both drives. If one drive fails, there is no data loss because the other drive contains the data. (Does not include disk striping.)	Redundancy. Better write performance than disk striping with parity.
0+1	Disk striping with mirroring	Two or more volumes, each on a separate drive, are striped and mirrored. Data is written sequentially to drives that are identically configured.	Redundancy with good read/write performance.
0	Disk striping	Two or more volumes, each on a separate drive, are configured as a stripe set. Data is broken into blocks, called stripes, and then written sequentially to all drives in the stripe set.	Speed/Performance without data protection.

Speed-up del modello base (IX)

- RAID5: Stripe set con parità
  - Si aumentano le prestazioni (4 scritture in parallelo)
  - Si aumenta l'affidalità (uno stripe per la parità)

Limiti allo speed-up nel modello base

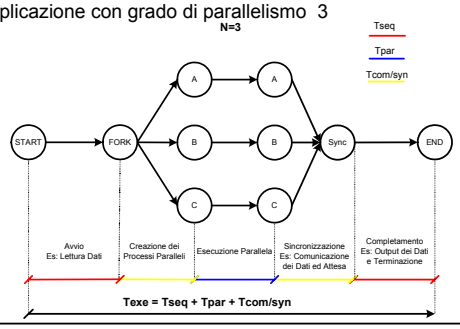
- Fino a che punto spingersi con tali ottimizzazioni?
  - **L'analisi di Tbusy rispetto alle altre componenti permette di stabilire il limite di tali tecniche**
  - Quando le altre componenti sono al di sotto del 10-5% di Tbusy, il sistema può considerarsi ottimizzato

Speed-up ulteriore (I)

- Come diminuire Tbusy?
  - Si cambia l'architettura interna del processore
    - Ma occorre modificare tutto
  - Si aumenta il clock
    - Ma aumenta Tidle in proporzione
    - L'architettura non è più ottimizzata
    - se overclocking off-the-shelf (illegale)
      - .... la stabilità?
  - **Si sfrutta il parallelismo dell'applicazione, se disponibile**
    - Si esegue l'applicazione su una macchina multiprocessore (NCPU>1)

Speed-up ulteriore (II)

- Parallelismo di una applicazione: un esempio
  - Applicazione con grado di parallelismo 3



Speed-up ulteriore (III)

- Texe: tempo di esecuzione
- Tseq: frazione relativa all'elaborazione sequenziale
  - Esempio:
    - attività di lettura dei dati ed inizializzazione dell'ambiente
    - scrittura dei dati sull'output e terminazione
- Tpar: frazione relativa all'elaborazione parallela
- Tcom/syn: frazione impegnata nella comunicazione/sincronizzazione fra i processi
  - Esempio:
    - Tempo per la creazione delle attività parallele
    - Tempo necessario allo scambio dei dati (overhead di comunicazione)
    - Tempo atteso perchè tutti i processi paralleli completino l'elaborazione (overhead di sincronizzazione)

Speed-up ulteriore (IV)

- Caso Ideale
  - Si assume che l'overhead Tcom/syn=0
    - creazione dei processi immediata
    - scambio dei dati immediato
    - sincronizzazioni immediate
  - Allora: se base è una macchina monoprocessore, l'applicazione ha grado di parallelismo N e la soluzione opt è una architettura ad N processori, si ha:

$$Texe_{base} = Tseq + Tpar \Rightarrow Texe_N = Tseq + Tpar / N$$

  - Lo si vede graficamente

Non conviene utilizzare una macchina con N1>N processori, altrimenti alcuni processori rimarrebbero inutilizzati

Legge di Amdahl

- E' una legge che descrive, in termini di speed-up, il vantaggio ottenibile da una ottimizzazione, in relazione alla durata della fase ottimizzabile.

$$Speedup_{opt} = \frac{Texe_{base}}{Texe_{opt}}$$

$$se\ Texe_{base} = Tseq + T_{opti} \Rightarrow Texe_{opt} = Texe_{base} * ((1-\alpha) + \alpha / Speedup_{opti})$$

$$con\ \alpha = Frazione\ Elab.ottimizzabile = T_{opti} / Texe_{base}$$

$$e$$

$$Speedup_{opti} = \text{Speedup frazione ottimizzabile} = T_{opti} / T_{opti1}$$

$$Speedup_{opt} = \frac{Texe_{base}}{Texe_{opt}} = \frac{1}{(1-\alpha) + \alpha / Speedup_{opti}}\ \text{Legge di Amdahl}$$

caso di elaborazione parallela (I)

$$Speedup_{opt} = \frac{Texe_{base}}{Texe_{opt}}$$

$$se\ Texe_{base} = Tseq + Tpar \Rightarrow Texe_N = Tseq + Tpar / N$$

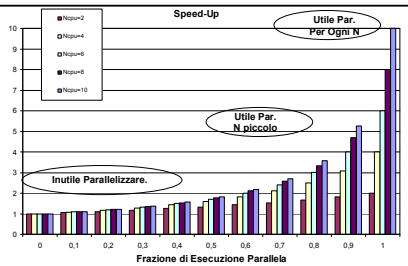
dove N è il numero di CPU

$$se\ \alpha = Frazione\ Elab.Parallela = Tpar / Texe_{base}$$

$$Speedup_N = \frac{N}{N(1-\alpha) + \alpha}\ \text{Legge di Amdahl}$$

caso di elaborazione parallela (II)

- Significato



  - Solo per alpha>0,9 ha senso parallelizzare (per N grande)
  - Per alpha piccoli, la maggior parte del guadagno si ottiene per N bassi

caso di elaborazione parallela (III)

- Estremi:
  - Frazione di esecuzione parallela = 0
    - Processi dipendenti
    - Al limite un solo processo
  - Frazione di esecuzione parallela = 1
    - Processi indipendenti
      - Es: N comandi Unix in esecuzione
      - Un server Web ed N utenti connessi che accedono alle pagine contemporaneamente
      - N query indipendenti in esecuzioni su di un DB
    - Frazione prossima ad uno:
      - DB con query non indipendenti, sistemi multithreaded
      - **Può modellare l'overhead di comunicazione/sincronizzazione**

caso di elaborazione parallela (IV)

- Considerazioni
  - Interpretazione:
    "l'architettura parallela conviene se si ha sufficiente parallelismo"
      - .....E tanto (alfa = .9)
  - ciò se si è interessati al rapporto prestazioni/prezzo
    - Vantaggio incrementale nell'aggiunta di un processore
  - Se le prestazioni sono un must, allora si dimensiona il sistema al massimo
    - per sistemi critici, è tollerabile spendere qualunque cifra
    - Anche con vantaggi incrementali bassissimi

Altra interpretazione della legge di Amdahl

$$Speedup_N = \frac{N}{N(1-\alpha)+\alpha} \text{ Legge di Amdahl (I)}$$

$$Speedup_N = \frac{1}{(1-\alpha)+\alpha/N} \text{ Legge di Amdahl (II)}$$

$$N = Tpar / Tenh = Speedup_{enhanced}$$

$$Speedup_{overall} = \frac{1}{(1-\alpha)+\alpha/Speedup_{enhanced}} \text{ Legge di Amdahl (II)}$$

Esempio - le FP/I

- In un benchmark, l'esecuzione della FP square root è responsabile del 20% del tempo di esecuzione. Tutte le istruzioni FP sono responsabili del 50% del tempo di esecuzione.
- Si possono effettuare 2 ottimizzazioni:
  - Aggiungere hw dedicato per accelerare di un fattore 10 la FP square root
  - Aggiungere hw dedicato per accelerare di un fattore 2 tutte le istruzioni FP
- Qual e' la più conveniente?

Esempio - le FP/II

- Applico l'enunciato II della legge di Amdahl

$$Speedup_{overall} = \frac{1}{(1-\alpha) + \alpha / Speedup_{enhanced}} \text{ Legge di Amdahl (II)}$$

$$Speedup_{FPSQR} = \frac{1}{(1-0.2) + 0.2/10} = \frac{1}{0.802} = 1.25$$

$$Speedup_{fp} = \frac{1}{(1-0.5) + 0.5/2.0} = \frac{1}{0.75} = 1.33$$

- E' più conveniente la seconda ottimizzazione

Esempio – i vantaggi delle cache

- Supponiamo di avere un sistema con un solo livello di cache ed i timing della slide 3 (ossia cache 10 volte più veloce della memoria)
- Supponiamo che i dati sono reperiti in cache il 90% delle volte (hit)
  - Valore realistico, anzi basso
- Qual è lo speed-up fra un sistema con cache ed uno senza cache?

$$Speedup_{overall} = \frac{1}{(1-\alpha) + \alpha / Speedup_{enhanced}} \text{ Legge di Amdahl (II)}$$

$$Speedup_{FPSQR} = \frac{1}{(1-0.9) + 0.9/10} = \frac{1}{0.19} = 5.3$$

- **Tale risultato giustifica l'adozione delle cache**

Architetture attuali (I)

- Architetture Parallele “Attuali”
  - commerciali (per le applicazioni oggi in voga)
  - ma anche di ricerca (...e molto)
  - Architetture Multiprocessore:
    - UMA / SMP
    - NUMA
  - Altre soluzioni per processi indipendenti
    - Clustering (sottoinsieme dei sistemi distribuiti)
      - Scalabilità
      - Alte prestazioni
      - Elevata Affidabilità

Architetture attuali (II)

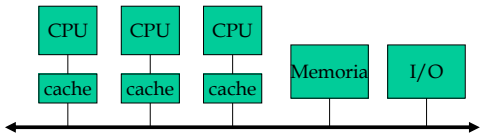
- Avvertenza prestazioni/affidabilità:
  - Non confondere prestazioni (velocità qui) con affidabilità
  - Multiprocessori:
    - Possono essere progettati per le prestazioni
      - Ma l’affidabilità diminuisce
        - » Ho più elementi
    - Possono essere progettati per l’affidabilità
      - Sistemi ridondati
      - Ma le prestazioni diminuiscono e/o il prezzo aumenta
        - » Occorre garantire la resistenza ai guasti
        - » Le spese includono il controllo della qualità (se non è passato, occorre buttare il pezzo)
  - Cluster:
    - Quelli per le prestazioni differiscono da quelli per l’affidabilità

Architetture attuali (III)

- Avvertenza cluster/sistema distribuito
  - Un cluster è un sistema distribuito
  - Rispetto ad altre applicazioni distribuite, cambia la granularità
    - A livello del S.O. per il cluster
    - A livello dell’applicazione per altre applicazioni distribuite
      - Es: Applicazione CORBA e CLUSTER

Sistemi SMP (I)

- SMP: sistema a processori simmetrici
  - ogni processore è equivalente agli altri per l'accesso alla memoria
  - i sistemi shared-bus shared-memory rientrano in questa categoria
    - Sono la "naturale" estensione off-the-shelf dei sistemi monoprocessore



Sistemi SMP (II)

- I principali sistemi operativi supportano tali architetture
  - Unix/Linux/Windows NT/2000/2003
  - Non richiedono eccessivi sforzi di porting
  - …ma attenzione alle lock!!!
- L'incremento delle prestazioni si ottiene aggiungendo CPU alla scheda
  - Le main-board sono simili a quelle monoprocessore
    - Slot vuoti per le CPU
  - In alcuni sistemi avviene "a caldo"
- Rappresentano l'architettura base di molti sistemi ad alte prestazioni
- Ampiamente utilizzati per web-server ad altissime prestazioni e per DB di medie dimensioni ad alte prestazioni

Sistemi SMP (III)

- alcuni esempi
  - Il sistema di fatturazione di TIM utilizza come nodo la SUN ENTERPRISE 10000 4way-SMP
  - Il sistema di autenticazione di Diners Europe utilizza le stesse macchine
    - E non sono delle "bombe"
      - Le architetture SUN sono affidabili, non veloci
  - Le "bombe":
    - Pentium IV costituisce il processore dei sistemi Compaq Professional Workstations 5100, 6000 ed 8000
      - fino a 4 -way SMP
      - architettura shared-bus shared memory, estensione di un PC classico (bus PCI-ISA, etc)

Sistemi SMP (IV)

- Le bombe: SPEC2000 Rate
  - Dati del 2001/2002

Company Name	System Name	#CPU	Base	Peak
Compaq	AlphaServer DS20E Model 6/667	1	424	444
IBM	RS/6000 SP-375MHz T/W	1	248	260
Intel	Intel VC820 (800 MHz Pentium III)	1	352	355
Intel	Intel VC820 (1.0 GHz Pentium III)	1	407	410
SGI	SGI 2200 2X 300MHz R12k	2	254	264
Sun	Ultra 10 333Mhz	1	133	-

Sistemi SMP (V)

- Alpha 21264 / 700 Mhz costituisce la base dei sistemi Compaq Alphaserver ES 40 e GS 140
  - fino a 14 processori su 7 nodi, ognuno biprocessore
  - un bus comune interconnette i sistemi alla memoria

- Limiti dei sistemi a bus condiviso
  - Il bus è un collo di bottiglia
  - Occorre mantenere coerenti le cache
    - Aumenta il traffico di bus
  - Sono Scalabili fino ad 8 processori
    - Lo speed-up è minimo oltre 8 CPU
    - Il bus è prossimo alla saturazione

Sistemi SMP (VI)

- Soluzione:
  - Si complica la rete di interconnessione
    - Diventa un crossbar switch (per rimanere un SMP)
      - Costoso ma scalabile (fino a 64/128 CPU)
      - **Consente più attivare più canali di comunazione contemporanei**
        - » Se sono coinvolte risorse diverse
    - Diventa una rete effettiva
      - Architettura NUMA (fino a 256 CPU ed oltre)
      - Il costo si sposta nel sw

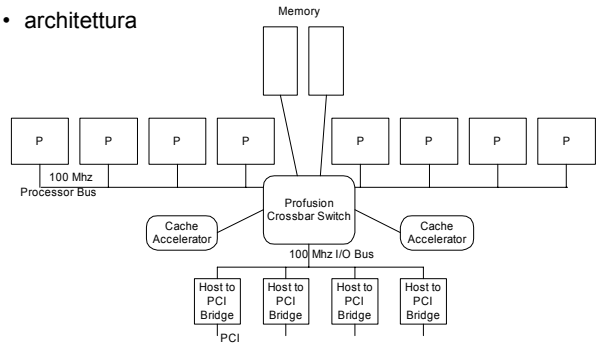
- Esempi:
  - power3 costituisce il processore adottato nel nodo 2 ways SMP alla base dell'architettura IBM RISC6000/SP
    - architettura NUMA, in cui ogni singolo nodo è un UMA a 2 processori
  - La famiglia di server Compaq ProLiant 8 ways SMP è basata sul processore PentiumIII
    - alcuni accorgimenti sono adottati per renderla UMA

Compaq 8-way SMP (I)

- Introduzione
  - il traffico di bus rende impossibile la connessione di più di 4 PentiumIII-IV su di un unico bus verso la memoria
  - per arrivare ad 8 processori, è stato sviluppato da Compaq un adeguato crossbar switch
  - altri accorgimenti sono stati adottati per limitare il traffico di bus indotto dal mantenimento della coerenza

Compaq 8-way SMP (II)

- architettura



Compaq 8-way SMP (III)

- **Il crossbar switch consente 10 accessi random contemporanei fra 10 porte (5 per la lettura e 5 per la scrittura)**
  - è non blocking, ossia permette contemporaneamente letture e scritture
- il bus a 100 MHz è il bus AGTL+ della Intel, che può reggere fino a 5 carichi.
  - Da qui lo schema di connessione (4 CPU - MEM oppure 4 PCI Bridge - MEM)
- i moduli cache accelerator servono per minimizzare il traffico di bus pur mantenendo la coerenza
  - un nodo tiene traccia di tutti i dati presenti nelle cache di un bus e sta in ascolto sull'altro bus, evitando di propagare le transazioni da un bus all'altro

Compaq 8-way SMP (IV)

- Caratteristiche
  - fino a 32G SDRAM
  - memoria 2 way interleaved
  - fino a 16 dispositivi PCI
  - Speed-up: 8 way raggiunge uno speed-up di 5 per il TPC-C
    - è 5 volte più rapido di un mono-processore per il benchmark TPC-C
    - La frazione di esecuzione parallela è di circa il 90%
- Nota: tali macchine sopravvivono ai processori

Architetture NUMA (I)

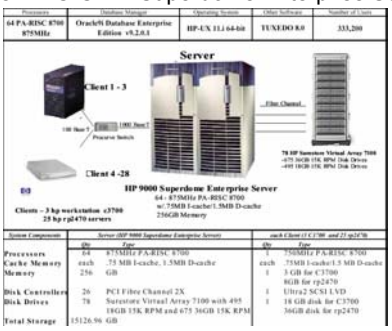
- NUMA
  - I processori non sono più simmetrici per i processi
  - E' importante tener traccia della CPU su cui è in esecuzione il processo
  - Lo scheduling è complicato
    - I sistemi operativi non sono più semplici
  - Esempio:
    - Windows 2000 Datacentric Server supporta architettura NUMA
    - …ma si compra preinstallato
    - E' sviluppato insieme da Microsoft e dai costruttori dell'hw
    - **Quanto costa? 56.000 $**

Architetture NUMA (II)

- Applicazioni
  - DB per sistemi OLTP e DSS di grandi/grandissime dimensioni
  - Applicazioni datacentric/Ntiered
- Hanno costo elevato
  - Sia HW che SW
- Esempio
  - Esempio: SGI Origin fino a 128 processori
    - Per OLTP e Grafica
  - IBM enterprise Server S80
    - 24 CPU
  - HP superdome
    - 64 CPU

Architetture NUMA (III)

- Esempio: per TPC-C HP Superdome Enterprise Server

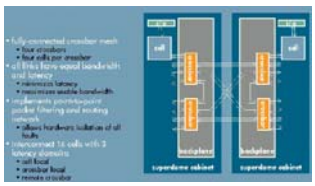


Architetture NUMA (IV)

- Caratteristiche del server
  - 64 CPU
  - 332K utenti
  - 256G byte RAM
  - 15 T byte di memorizzazione
  - Server
    - Architettura CC-NUMA a maglia (mesh)
    - Il singolo nodo è 4-way SMP
    - Protocollo di coerenza basato su directory
    - Mesh realizzata tramite crossbar

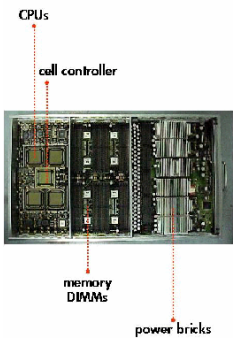
Architetture NUMA (V)

- Struttura della rete di interconnessione
  - 2 cabinet
  - 4 celle per crossbar
  - 4 crossbar interamente connessi



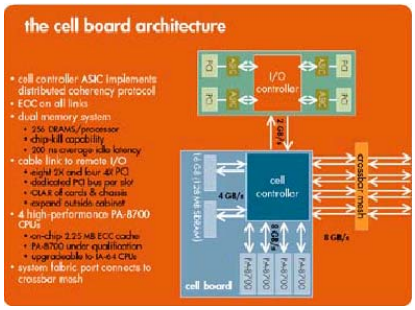
Architetture NUMA (VI)

– Struttura della cella
  • 4 CPU
  • RAM (DIMM)
  • 12 PCI I/O SLOT
  • CELL CONTROLLER (ASIC)



Architetture NUMA (VII)

– Architettura complessiva e banda



Architetture NUMA (VIII)

– Uno sguardo ai prezzi
  • Sistema complessivo: 5,8M $ + 750K $ per manutenzione
    – I client (28) rientrano nel prezzo
      » È un OLTP: anche i client sono elementi del sistema e partecipano alle prestazioni
  • SERVER HW:
    – RAM: 1,7M $
    – CPU: 161K $ + 1,4 M $ Rigth
    – Cabinet (chassis) per crossbar: 500K $
  • **SERVER SW:**
    – Oracle 9i: **1,2 M $**
    – Ma è gratis (lo stesso) su windows2000

Architetture NUMA (IX)

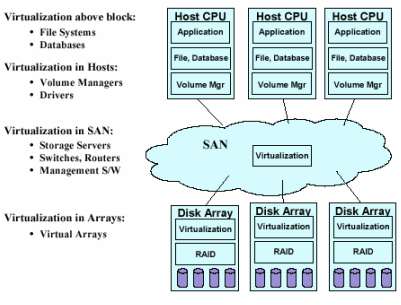
- Server Storage
  - 78 hp surestore **virtual array** controller: **3,4M** $
  - Oltre 1000 hd di vario formato: **1,3M $**
  - **Nota: il costo totale del sistema è di 5 milioni perchè prevede uno sconto di 5M $**
    - **Il costo aggiuntivo di manutenzione è di 743K $**
- **Virtual Array: vedi dopo**

SAN (I)

- Ancora sullo storage: SAN e virtualizzazione
  - RAID è un livello di virtualizzazione del disco
  - Ad esso se ne aggiungono altri:
    - SAN: Storage Area Network
      - Una rete di dispositivi di memorizzazione controllati da una o più CPU (il "client" del disco, ossia un DB server, un file server, etc
      - La condivisione è a livello del blocco
      - Si "virtualizza" tale rete
        - » **Se ne maschera la struttura fisica**
    - virtual array
      - Un array di dschi con funzioni di virtualizzazione superiore al RAID

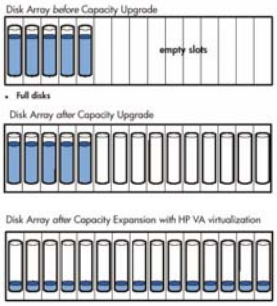
SAN (II)

- Il legame fra i vari livelli di virtualizzazione



SAN (III)

- Esempio di funzione di virtualizzazione
- Array pieno
- Upgrade:
  - Dischi nuovi vuoti
  - Dischi vecchi pieni
  - Bilanciamento a mano
- Upgrade con dischi virtuali:
  - Bilanciamento automatico



I Cluster (I)

- Definizione
  - Un insieme di nodi visti, per quel particolare servizio o applicazione, come un solo nodo
    - "purtroppo" devono essere omogenei
- Applicabilità
  - Processi indipendenti
    - Ideali per applicazioni N-Tiered
- Vantaggi:
  - Scalabilità
  - Alte prestazioni
  - Affidabiità
  - Costo
    - A parità di #CPU, sono più economici deo CC-NUMA

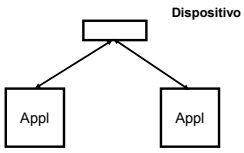
I Cluster (II)

- Tassonomia (I)
  - Shared nothing:
    - Ideali per scalabilità e bilanciamento del carico
    - Es: web server
  - Shared-resource
    - Per soluzioni affidabili
    - Aggiornano le risorse condivise (es: DB)

Shared-Nothing **Shared-resource**

I Cluster (III)

- Tassonomia (II)
  - **Cluster based**
    - È visibile un indirizzo IP
    - Corrisponde ad un dispositivo
    - Tale dispositivo effettua lo scheduling delle richieste
    - Ideale per il Load-Balancing



I Cluster (IV)

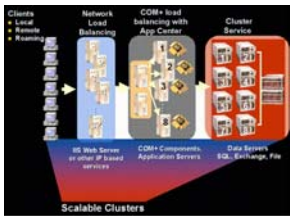
- **Virtual Cluster**
  - Non è presente il dispositivo che esporta l'indirizzo IP
  - I nodi decidono se prelevare o no il pacchetto
    - Hanno lo stesso indirizzo IP
    - Si scambiano informazioni "sullo stato di salute"
    - Un nodo fault non riceve più i pacchetti
  - Ideale per il load-balancing ed il recovery
- **Distributed Cluster**
  - La risoluzione avviene a livello del DNS, ossia il client invia un'unica risposta
  - E' il DNS che effettua lo scheduling
  - Valido per cluster "distribuiti"

I Cluster (V)

- Esempio:
  - Microsoft WINDOWS 2000 Advanced server fornisce 3 livelli di clustering:
    - NLB virtuale, shared nothing
      - Ideale per IIS
      - Per soluzoni scalabili e bilanciate
    - CLB (Component Load Balancing) virtuale, shared nothing
      - Per i componenti DCOM e .net
      - Per soluzioni scalabili e bilanciate
    - Server Cluster virtuale, shared resource
      - Per SQL Server
      - Per soluzioni ad elevata affidabilità

I Cluster (VI)

- Possono essere combinati insieme
- Esempio di applicazione 3-tiered clustered ai tre livelli



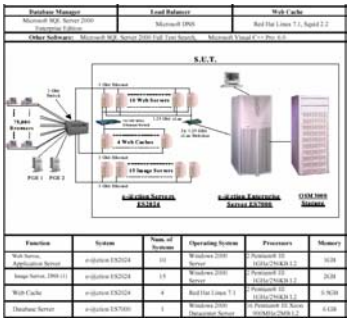
I Cluster (VII)

- Capacità

Operating System	Edition	Network Load Balancing	Component Load Balancing	Server Cluster
Windows 2000				
	Advanced Server	32	8	2
	Datacenter Server	32	8	4
Windows .NET				
	Advanced Server	32	8	4
	Datacenter Server	32	8	8

I Cluster (VIII)

- Esempio: TPC-W per Unisys E-@ction Enterprise Server ES7000



I Cluster (IX)

- TPC-W è un benchmark per sistemi 3-tiered
  - Specifico per E-Commerce
- L'architettura è di cluster basati sul DNS, di tipo load-balancer
  - Eccetto il DB che è singolo
- Caratteristiche:
  - Architettura generale: WINDOWS 2000
    - Eccetto la web-cache su Linux
  - SWEB: 10 biprocessori
  - Image Server: 15 biprocessori
  - Web cache: 4 biprocessori
  - DB-Serve: 1 Server a 16 CPU SMP
    - Basato su crossbar switch switch

I Cluster (X)

- Uno sguardo ai prezzi
  - Complessivo: 1M $
    - I client non sono inclusi nel sistema
    - Per 75000 utenti
  - Web server HW+SW: 80K $
  - Image Server HW+SW: 133K $
  - WebCache HW+SW: 23K $
  - DB Server HW+SW: **670K $**
    - **192K $ per CPU**
    - **43K $ per RAM**
  - DB Storage:**113K $**
  - Network HW: 11K $

I Cluster (XI)

- Confronti di prezzo:
  - Il costo dei crossbar switch rispetto ad SMP:
    - 1 ES7000, 0 CPU, 0 RAM: 90K $ (la board con switch)
      - Subpoad 4 900Mhz Xeon: 48 K $ (il nodo 4-way SMP)
    - 1 ES2024, 0 CPU, 0 RAM: 1,5 K $ (la board 2way SMP)
  - Sistemi Operativi
    - Windows 2000 SMP (base): 0,738 K $
    - Windows 2000 Datacenter Server (per ES7000): **50K** $
  - DB
    - SQL server 2000:
      - Licenza per CPU: 16K $
      - Prezzo licenze totali: **265K $**

I Cluster (XII)

- L'incidenza delle manutenzione (3 anni):
  - Web: 10 sistemi 13K $
  - Image: 15 sistemi 19K $
  - Cache: 4 sistemi 22K $
    - Il supporto red-hat per 3 anni incide per 17K $
  - DB: 1 sistema 108K $
  - DB storage hw: 6K $
  - Network: 2,3 K $

ANCORA??????