Arquitecturas Paralelas e Distribuídas

Luís Nogueira

luis@dei.isep.ipp.pt

Departamento Engenharia Informática

Instituto Superior de Engenharia do Porto

Motivação

- Criar computadores poderosos ligando vários CPUs
  - Custos de desenvolvimento de novos processadores
  - Limites da tecnologia de semicondutores
- Diminuir tempo de execução dos programas
  - Aumentando nº instruções debitadas
- Tratar problemas mais complexos
  - Genética, farmacologia, física, . . .
- Sistemas robustos e tolerantes a falhas

Limites do aumento de performance

- Aumento de performance (speedup) não é linear
- Um programa é constituído por
  - Fracção executada sequencialmente (s)
  - Fracção que pode ser executada em paralelo (p)
- Tempo de execução de um programa
  - $s + p = 1$

$$Speedup = \frac{\text{tempo execução 1 processador}}{\text{tempo execução n processadores}} = \frac{T(1)}{T(n)}$$

Aumento de performance - Lei de Amdhal

$$Speedup = \frac{s+p}{s+\frac{p}{n}} = \frac{1}{s+\frac{1-s}{n}} = \frac{n}{ns+(1-s)}$$

- $s = 0 \rightarrow speedup = n$ (valor óptimo)
- $s = 1 \rightarrow speedup = 1$ (pior caso, sequencial)
- $s = 0.05, n = 10 \rightarrow speedup = 6.89$ (valor ideal era 10!)
- Se uma pequena parte do programa não poder ser executada em paralelo
  - Enorme impacto no hipotético aumento de performance
- Speedup obtido é muito sensível a s

Aumento de performance - Lei de Gustafson-Barsis

- Gustafson e Barsis defendem que p e n dependem um do outro
  - Para usar mais processadores temos de aumentar tamanho do problema executado em paralelo

$$Speedup = n - (n-1)s$$

- $s = 0 \rightarrow speedup = n$ (valor óptimo)
- $s = 1 \rightarrow speedup = 1$ (pior caso, sequencial)
- $s = 0.05, n = 10 \rightarrow speedup = 9.55$ (mais optimista!)

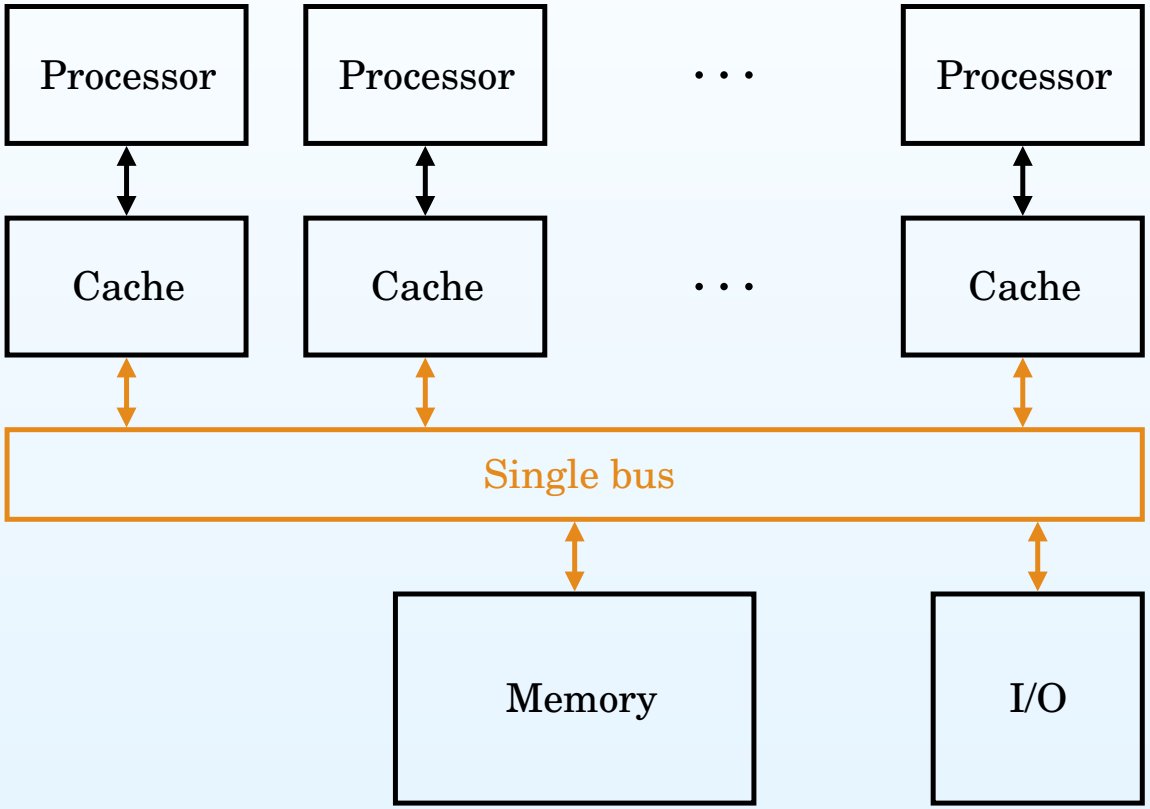
Questões

- Como partilhar dados pelos processadores?
  - Memória partilhada
  - Passagem de mensagens
- Como coordenar os diversos processadores?
  - Sincronização (semáforos)
  - Primitivas send/receive
  - Protocolos do S.O.
- Como ligar os diversos processadores?
  - Bus único
  - Rede de interconexão

Symmetric Multiprocessors (SMPs)

- Espaço de endereçamento físico global
  - Fisicamente partilhado através de um bus único
  - Latência no acesso à memória independente do processador
- Processadores comunicam através de variáveis partilhadas em memória
  - Sincronização através de semáforos
  - Modelo de programação atractivo
- Cada processador utiliza uma cache individual
  - Diminuir tráfego no bus partilhado
  - Mantém cópia de dados partilhados em memória

Symmetric Multiprocessors (SMPs)



Problema de coerência das caches em SMPs

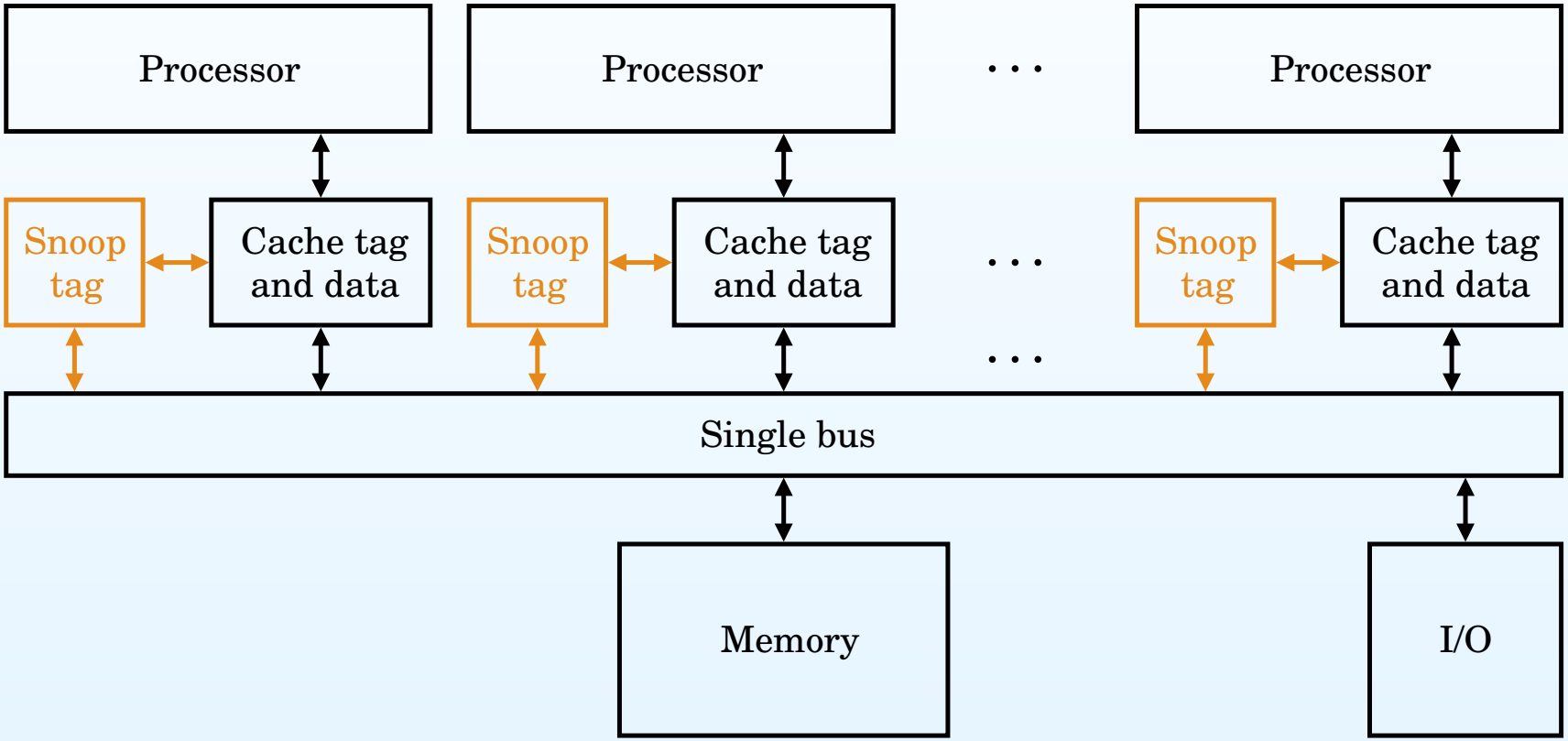
Tempo	Evento	Cache CPU A	Cache CPU B	Memória
0				1
1	CPU A lê X	1		1
2	CPU B lê X	1	1	1
3	CPU A escreve 0 em X	0	1	0

- Réplicas nas caches privadas têm de ser coerentes
- Necessário garantir
  - Escritas em memória são visíveis a todos os processadores (write propagation)
  - Ordem das escritas é a mesma para todos os processadores (write serialization)

Snooping

- Controlador da cache monitoriza transacções que passam no bus
- Todas as transacções são visíveis a todos os controladores
- Todos os controladores veêm transacções pela mesma ordem
- Controladores actuam sobre blocos locais quando necessário
- Etiquetas dos blocos em cache são duplicadas
  - Evitar interferências quando CPU acede à cache
  - Controlador acede a etiquetas duplicadas

Snooping


Processor
Processor
Processor
Snoop tag
Cache tag and data
Snoop tag
Cache tag and data
Snoop tag
Cache tag and data
Single bus
Memory
I/O

Protocolos base de coerência de caches

- Write-update
  - Alteração num bloco na cache local implica actualizar cópias nas restantes caches
  - Múltiplas escritas no mesmo bloco $\longrightarrow$ múltiplas actualizações
- Write-invalidate
  - Alteração num bloco na cache local invalida cópias nas restantes caches
  - Múltiplas escritas no mesmo bloco $\longrightarrow$ apenas uma invalidação

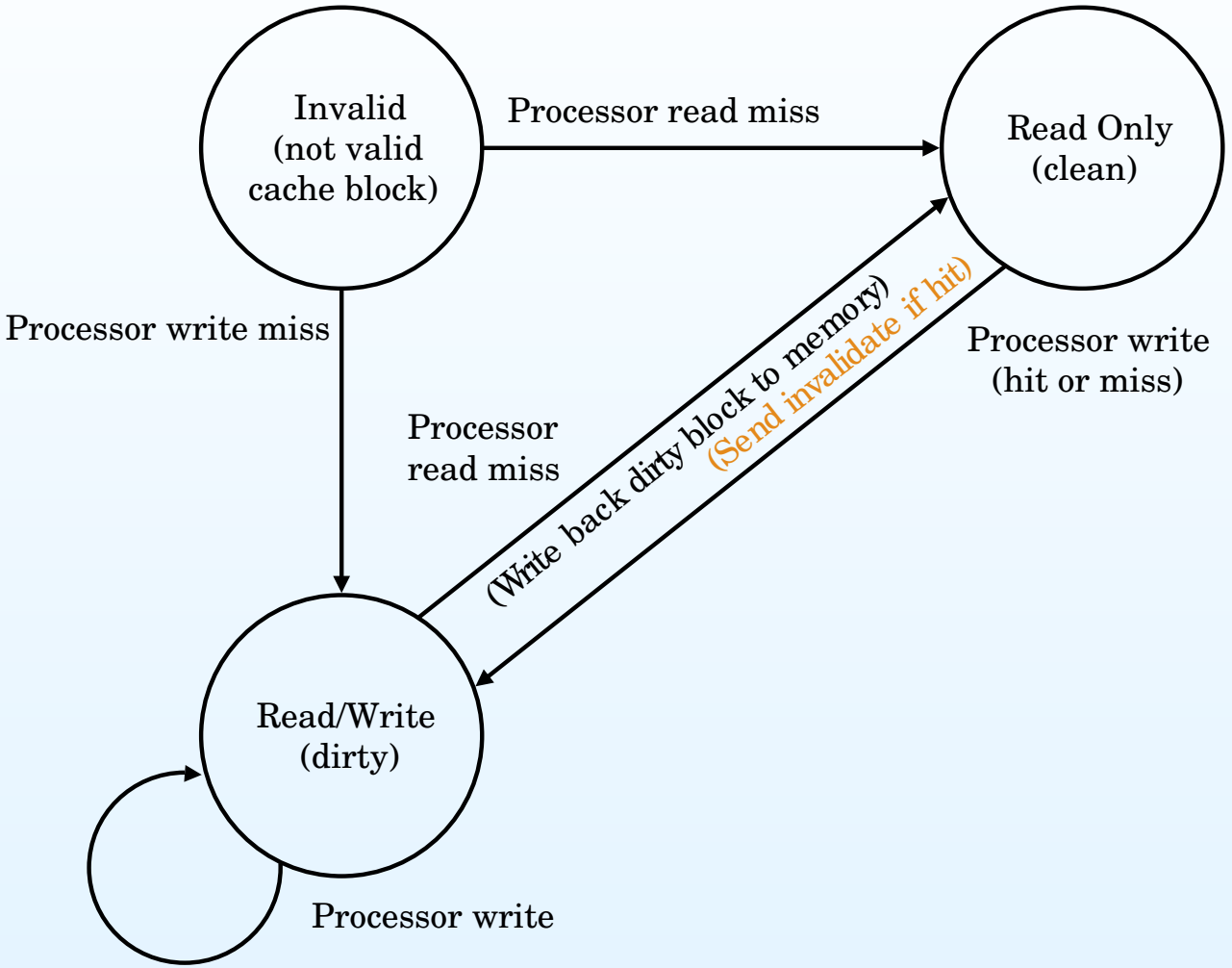
Tipos de cache

- Write-through
  - Bloco actualizado na cache privada e na memória partilhada
  - Coerência facilmente implementada em hardware
  - Gera mais tráfego no bus $\rightarrow$ problemático em SMPs
- Write-back
  - Bloco actualizado apenas na cache privada
  - Dirty bit minimiza escritas em memória
  - Coerência exige protocolos mais complexos
  - Gera menos tráfego no bus

Protocolo MSI com write invalidate/write back

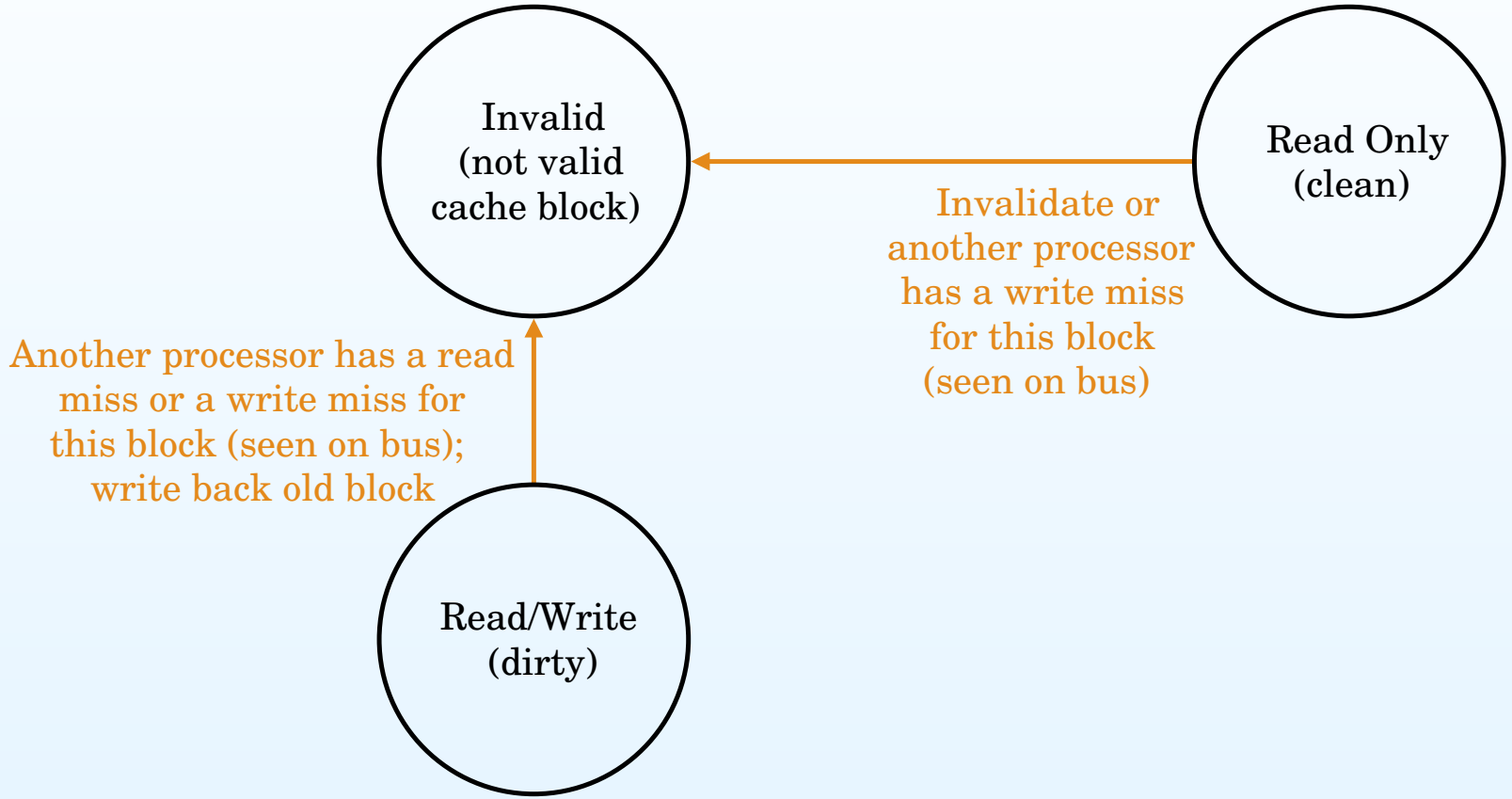
- Bloco da cache pode estar num de 3 estados (MSI)
  - Modified - bloco foi alterado e não pode ser partilhado
  - Shared - bloco está limpo e pode ser partilhado
  - Invalid - bloco não possui dados válidos
- Transições entre estados geradas pelo processador ou bus
  - Em read misses, write misses e write hits
  - Read hits não provocam mudança de estado
- Pentium 4 (e outros) acrescenta estado *exclusive* (MESI)
  - Não existem cópias do bloco
  - Não é necessário invalidar cópias num *write hit*

Protocolo MSI (bloco local)



a. Cache state transitions using signals from the processor

Protocolo MSI (cópias)



b. Cache state transitions using signals from the bus

Sincronização do acesso à memória

- Necessário coordenar processadores que acedem a variáveis partilhadas
  - Conflito resolvido com semáforo sobre bus partilhado
- Apenas um processador tem acesso ao bus num dado momento
  - Acessos sequenciais à memória
- Garante que
  - Escritas em memória são visíveis a todos os processadores (write propagation)
  - Ordem das escritas é a mesma para todos os processadores (write serialization)

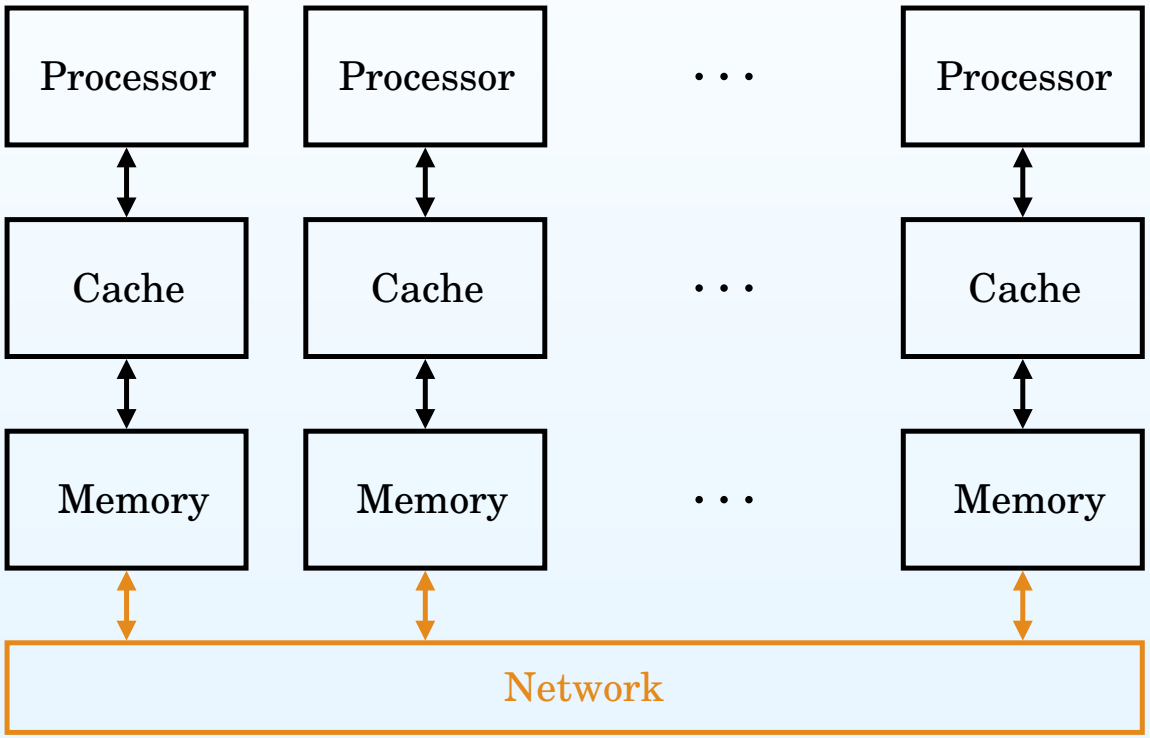
Limitações dos SMPs

- SMPs limitados a um nº relativamente pequeno de processadores
  - Até à data o máximo conseguido foram 36
- Limites do bus partilhado
  - Elevada largura de banda, baixa latência e grande comprimento do bus são incompatíveis
- Limites na largura de banda da memória
- Limites impostos pelo modelo de sincronização

Processadores ligados por rede de interconecção

- Cada processador possui memória e caches privadas
- Processadores ligados por rede de interconecção
  - Especializada e com diversas topologias
- Relativamente à memória
  - Distribuída
    - Memória fisicamente distribuída pelos processadores
  - Virtualmente partilhada
    - Memória fisicamente distribuída mas logicamente partilhada

Processadores ligados por rede de interconecção



Topologias das redes de interconecção

- Redes estáticas
  - Ligações entre CPUs são fixas
  - Mensagens seguem rotas bem definidas
- Redes dinâmicas
  - Usam switches nas ligações entre CPUs
  - Caminhos para as mensagens encontrados dinâmicamente
- Vamos analisar as características
  - Largura de banda, latência, custo
  - Nº máximo de hops entre CPUs, conectividade

Redes de interconecção estáticas

- Estrela
  - Todas as mensagens passam por um nó central
  - Necessita $n - 1$ ligações
  - Nº máximo de hops: 2
  - Apenas um caminho possível para uma mensagem
  - Rede desconexa se falhar apenas 1 ligação
- Anel
  - Nós sequencialmente ligados (incluindo extremidades)
  - Necessita n ligações
  - Nº máximo de hops: $n/2$
  - Dois caminhos alternativos para uma mensagem
  - Rede desconexa se falharem 2 ligações

Redes de interconecção estáticas

- Completamente ligadas
  - Ligações directas entre todos os CPUs
  - Necessita $n(n-1)/2$ ligações
  - Nº máximo de hops: 1
  - $n-1$ caminhos possíveis para uma mensagem
  - Rede desconexa se falharem $n-1$ ligações
- Redes completamente ligadas são muito caras
- Necessário encontrar equilíbrio entre
  - Custo das redes em anel
  - Performance das redes completamente ligadas

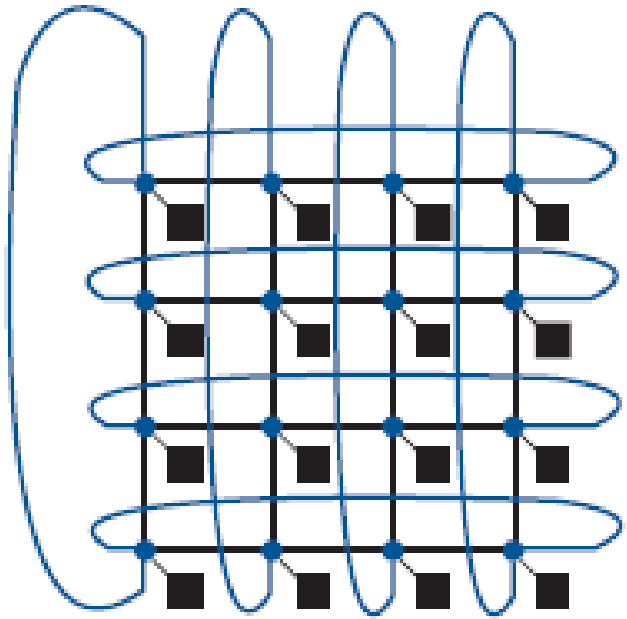
Redes de interconecção estáticas

- Grelha 2-D
  - CPUs ligados por sistema de coordenadas linha/coluna
  - Melhor performance e tolerância a falhas (Torus 2D)
    - As extremidades são também ligadas, criando grelha de anéis horizontais e verticais
  - Necessita $2 * n$ ligações
  - Nº máximo de hops: $2 * \lfloor \sqrt{n/2} \rfloor$
  - 4 caminhos possíveis para uma mensagem
  - Rede desconexa se falharem 4 ligações

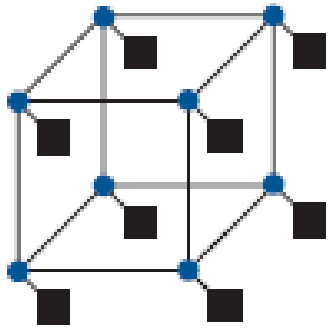
Redes de interconecção estáticas

- Hipercubo
  - Cubo binário n-dimensional (2^n CPUs)
  - Cada CPU está ligado a n vizinhos
  - Necessita $(n * \log_2 n)/2$ ligações
  - Nº máximo de hops: $\log_2 n$
  - $\log_2 n$ caminhos possíveis para uma mensagem
  - Rede desconexa se falharem $\log_2 n$ ligações

Redes de interconecção estáticas



a. 2-D grid or mesh of 16 nodes

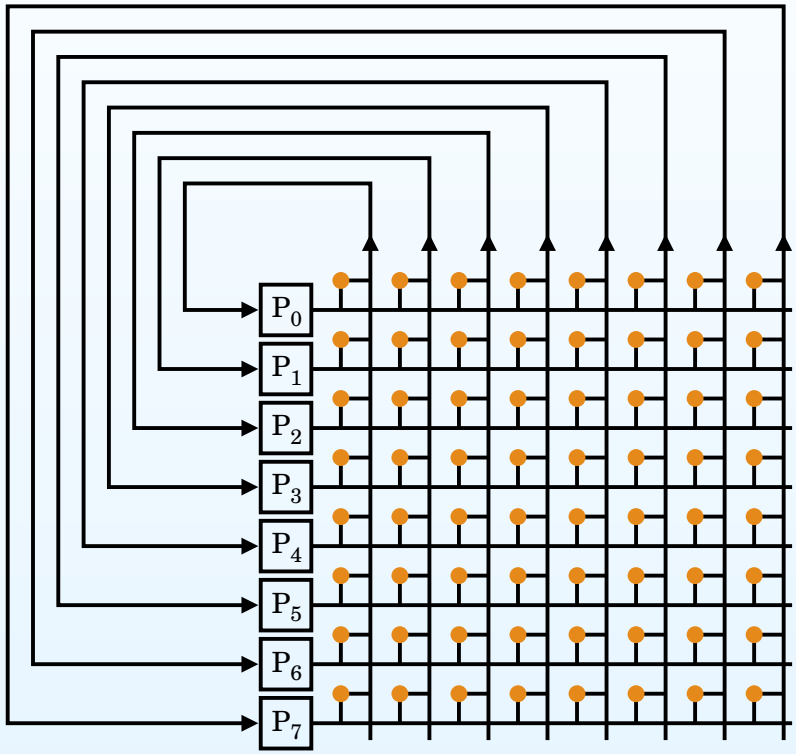


b. n-cube tree of 8 nodes ($8 = 2^3$ so $n = 3$)

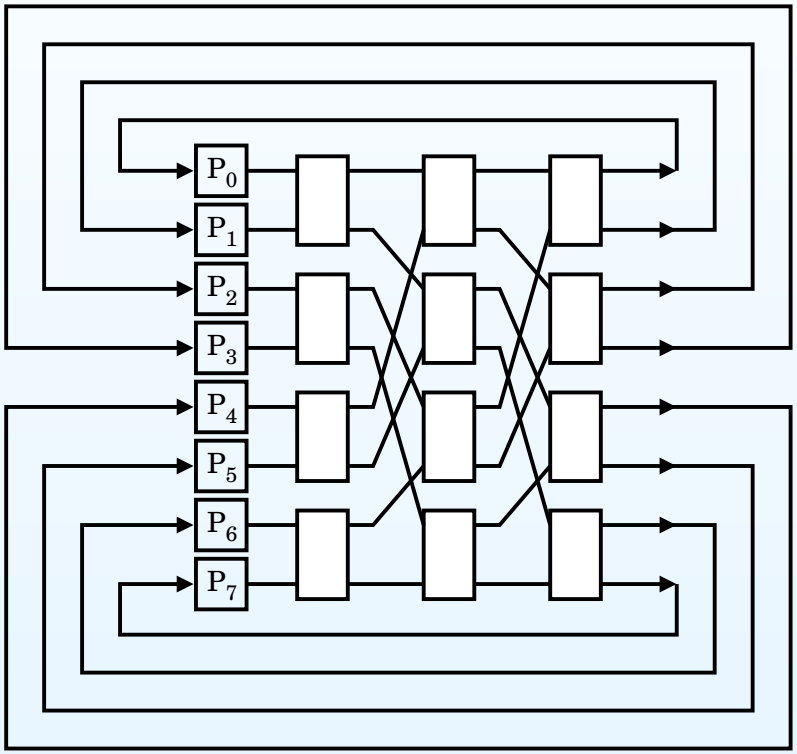
Redes de interconecção dinâmicas

- Denominadas redes multi-stage
  - Múltiplos caminhos possíveis para uma mensagem
- Crossbar
  - Rede completamente ligada através de grelha de switches binários
  - Usa n^2 switches
- Omega
  - Usa menos hardware ($2n \log_2 n$ switches)
  - Susceptível a congestionamento de tráfego

Redes de interconecção dinâmicas



a. Crossbar



b. Omega network

Arquitecturas de memória distribuída

- Memória fisicamente distribuída
  - Processador apenas endereça memória privada
- Exige distribuição de tarefas
  1. Distribuir explicitamente dados pelos n processadores
  2. Agrupar resultados
- Comunicação entre processadores é explícita
  - Enviar mensagem para partilhar dados
  - Receber mensagem para aceitar dados enviados

Arquitecturas de memória distribuída - Vantagens

- Arquitectura escalonável até grande nº de processadores
  - Denominada por Massively Parallel Processors (MPPs)
- Não existe bus partilhado
  - Máxima largura de banda para memória privada
  - Sem necessidade de sincronização do acesso à memória
- Não existem problemas de coerência nas caches
  - Caches possuem apenas blocos locais

Arquitecturas de memória distribuída - Desvantagens

- Comunicação entre processadores introduz overhead
  - Envio de dados exige envio de mensagem
  - Recepção de mensagens obriga interrupção do processador
- É mais difícil programar eficientemente
  - Necessário conhecer pormenores da arquitectura subjacente

Arquitecturas de memória virtualmente partilhada

- Memória fisicamente distribuída mas logicamente partilhada
  - Camada de software simula espaço endereçamento único
- Conceito semelhante a memória virtual em uniprocessador
  - Tabela de páginas indica se página local ou remota
  - Referências remotas são transformadas em mensagens
- Se distribuição dos dados pelos n processadores for aleatória
  - Performance de programas com elevado *miss rate* é muito má

Arquitecturas de memória virtualmente partilhada

- Performance muito dependente da localidade das referências
  - *Miss penalty* envolve mensagem pela rede
  - Largura de banda gasta com transferência de páginas
- Performance aumenta significativamente
  - Se o programador ou compilador conseguirem alocar dados ao processador que os irá usar
- Esta alocação não é tão complexa como em memória distribuída
  - Dados necessários podem sempre ser acedidos

Problema da coerência das caches

- Espaço de endereçamento único
  - Cópias do mesmo endereço em caches distintas
- Snooping não resolve problema
  - Não existe um bus partilhado onde são propagadas todas as referências à memória
  - Natureza distribuída do protocolo não é escalonável a um grande nº de processadores
  - Exige comunicação com todas as caches num *cache miss*
- Sem coerência das caches
  - Apenas dados privados podem estar na cache
  - Blocos partilhados são marcados como *uncacheable*

Protocolos de coerência baseados em directórios

- Directório central mantém informação dos blocos em cache
  - Quais as caches que possuem cópias do bloco
  - Estado do bloco
- Processador comunica alteração de bloco na cache ao directório
  - Directório invalida ou actualiza cópias
  - Não é necessário broadcast!
- Transições entre estados geradas por mensagens explícitas

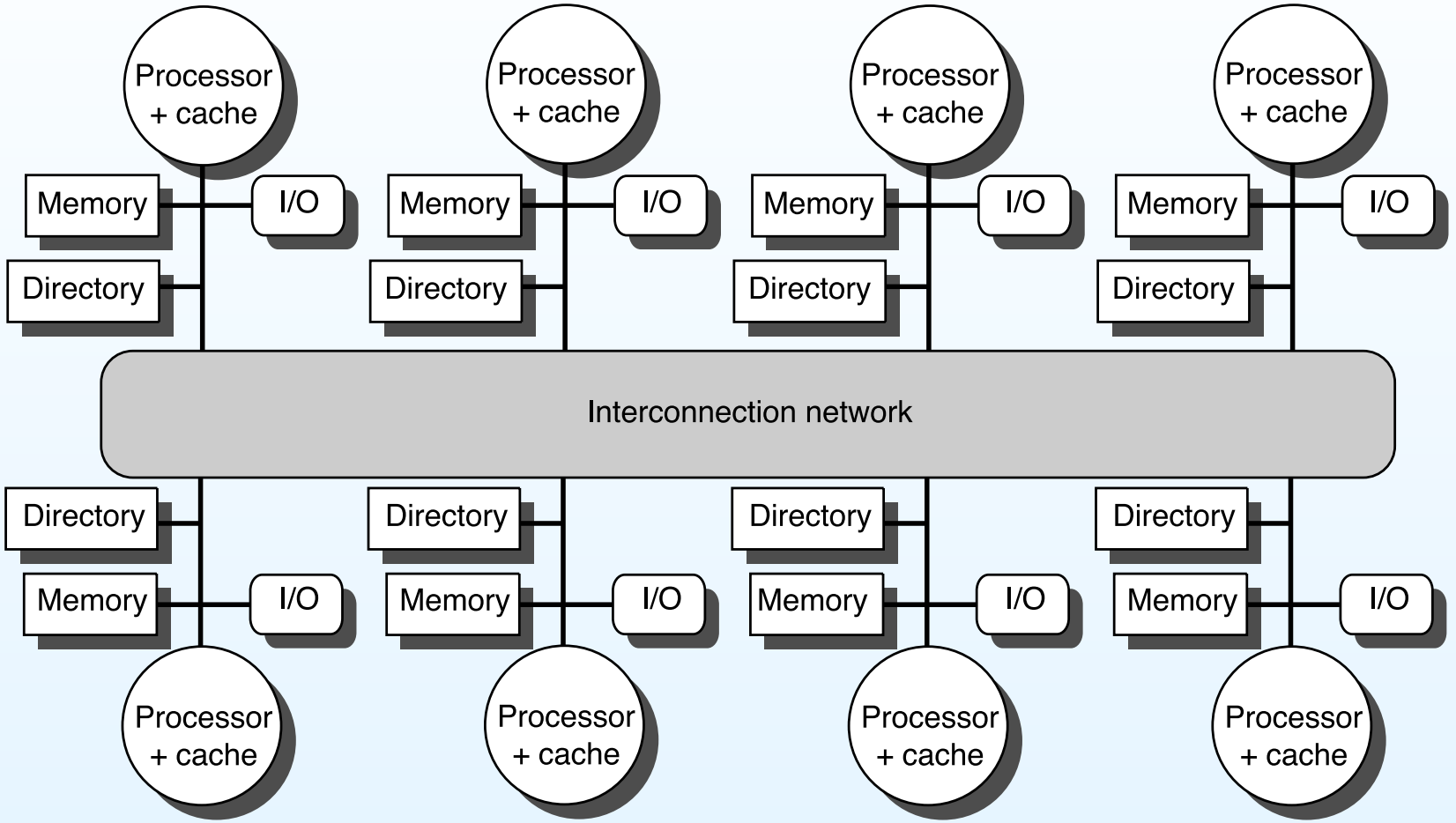
Protocolos de coerência baseados em directórios

- Protocolos tradicionais associam entrada no directório a todos os blocos em memória
  - Independentemente de serem ou não partilhados
  - Nº de entradas = nº de processadores * nº de blocos cache
  - Elevado consumo de memória em sistemas com muitos processadores (> 100)
- Maior nº de processadores exige protocolo que
  - Diminua nº de bits por entrada
  - Restrinja nº de entradas apenas aos blocos efectivamente partilhados

Distribuição de directórios

- Elevada escalabilidade com distribuição de directórios
  - Cada directório monitoriza caches que referenciam endereços físicos da porção de memória que controla
- Directórios distribuídos mantêm característica essencial
  - Informação sobre um dado bloco está apenas num único directório conhecido
  - Bits mais significativos usados como etiqueta

Distribuição de directórios



Clusters

- Essencialmente idênticos aos sistemas MPP
  - No entanto, cada nó é um PC (muitas vezes SMP)
  - Rede de interconecção é uma LAN implementada com Ethernet, ATM, FDDI, . . .
- Principais vantagens
  - Facilidade de expansão
  - Baixo custo dos PCs
  - Versatilidade
  - Melhor tolerância a falhas
- Principal desvantagem
  - Menor largura de banda e maior latência da rede de interconecção

Relação custo/performance

- Clusters (memória distribuída)
  - Bases de dados, servidores web/email, . . .
  - Baixo custo, boa performance, muito versátil
- Servidor multiprocessador (memória partilhada / virtualmente distribuída)
  - Bases de dados, sistemas de ficheiros, . . .
  - Custo intermédio, boa performance, optimizado para I/O
- Supercomputadores MPP
  - Cálculo numérico e simbólico
  - Custo muito elevado, elevada performance

Programas de avaliação de performance

- LINPACK (LInear PACK)
  - Resolução de sistemas de equações lineares por decomposição LU
  - Efectua $2n^3/3 + 2n^2$ operações (n - dimensão da matriz)
  - Performance indicada em TFLOP (Trillion Floating Point Instructions / Second)
- SPECrate
  - Mais genérico
  - Executa várias instâncias de programas intensivos em inteiros e vírgula flutuante
- LAPACK, Livermore Loops, NAS Kernel, PERFECT, SLALOM, . . .

TOP 500

- Consultar http://www.top500.org/
- Classificação baseada no LINPACK
- TOP 5 actual (Novembro 2006)
  1. BlueGene/L (IBM, EUA) - 280.6 TFLOPS (70.72 em 2004)
  2. Red Storm (NNSA, EUA) - 101.4 TFLOPs
  3. BGW (IBM, EUA) - 91.29 TFLOPs
  4. ASC Purple (DOE/NNSA/LLNL, EUA) - 75.76 TFLOPs
  5. MareNostrum (Barcelona SCC, Espanha) - 62.63 TFLOPs

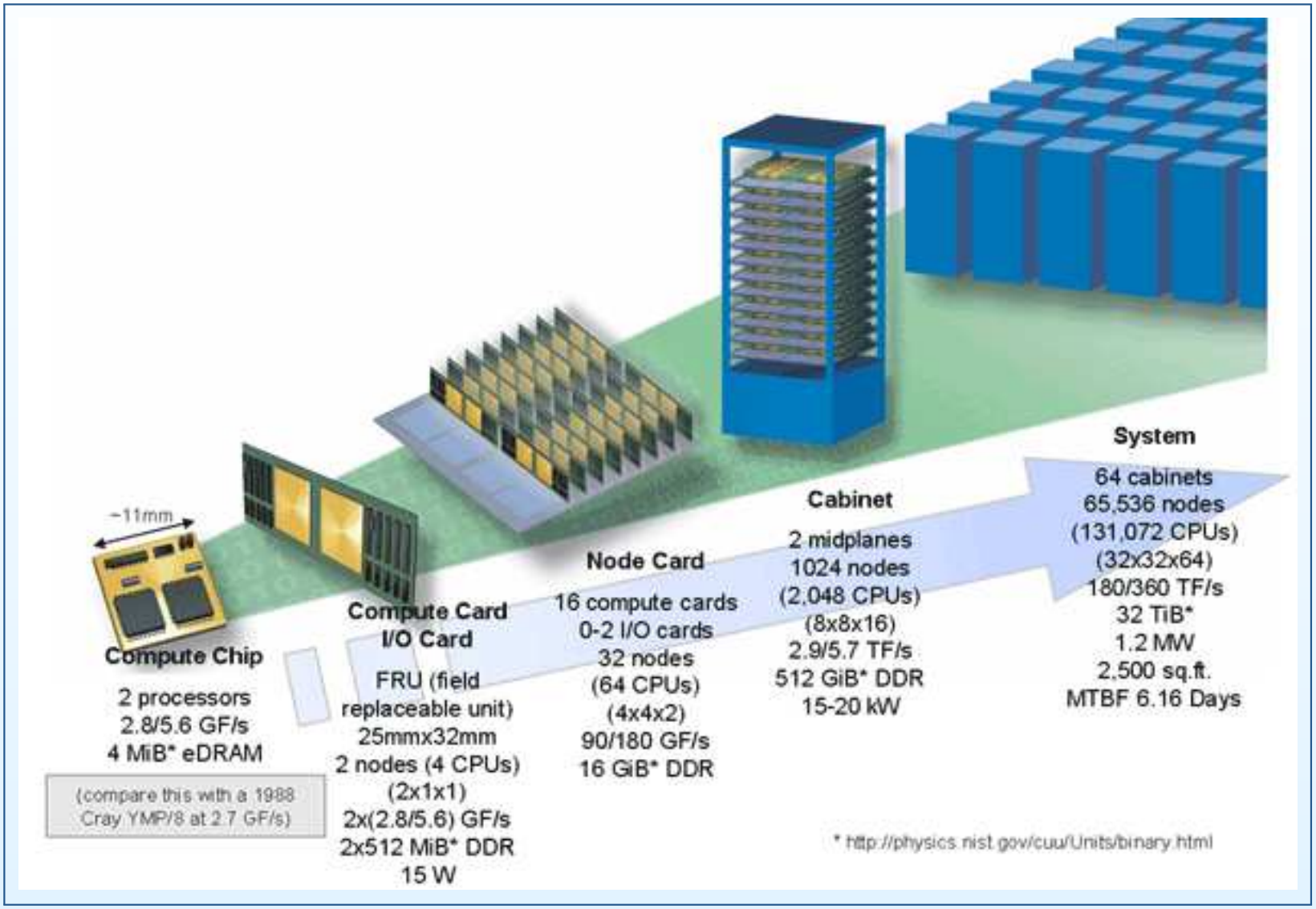
BlueGene/L

- MPP altamente escalonável
  - Objectivo final: 360 TFLOPs com 131072 CPUs
  - Actualmente possui 65536 CPUs
- Objectivos
  - Conseguir relação custo/performance atractiva
  - Diminuir necessidades de alimentação e arrefecimento
- Chip +/- 11mm (bloco básico da arquitectura)
  - 2 CPUs (0.7 GHz PowerPC 440)
    - Um para processamento e outro para comunicação
  - 4 MB Cache

BlueGene/L

- Placa de computação
  - 2 chips
  - 512 MB DDR SDRAM para cada chip
- Nós interligados por múltiplas redes de interconceção complementares
  - Elevada largura de banda e baixa latência
  - Rede Torus 3D para comunicação ponto a ponto
  - Rede em árvore para operações globais
  - Gigabit Ethernet para comunicação com outros sistemas

BlueGene/L


~11mm
Compute Chip
2 processors
2.8/5.6 GF/s
4 MiB* eDRAM
(compare this with a 1988 Cray YMP/8 at 2.7 GF/s)
Compute Card
I/O Card
FRU (field replaceable unit)
25mmx32mm
2 nodes (4 CPUs)
(2x1x1)
2x(2.8/5.6) GF/s
2x512 MiB* DDR
15 W
Node Card
16 compute cards
0-2 I/O cards
32 nodes
(64 CPUs)
(4x4x2)
90/180 GF/s
16 GiB* DDR
Cabinet
2 midplanes
1024 nodes
(2,048 CPUs)
(8x8x16)
2.9/5.7 TF/s
512 GiB* DDR
15-20 kW
System
64 cabinets
65,536 nodes
(131,072 CPUs)
(32x32x64)
180/360 TF/s
32 TiB*
1.2 MW
2,500 sq.ft.
MTBF 6.16 Days
* http://physics.nist.gov/cuu/Units/binary.html