

# Parallele Rechnerarchitektur II

Stefan Lang

Interdisziplinäres Zentrum für Wissenschaftliches Rechnen  
Universität Heidelberg  
INF 368, Raum 532  
D-69120 Heidelberg  
phone: 06221/54-8264  
email: [Stefan.Lang@iwr.uni-heidelberg.de](mailto:Stefan.Lang@iwr.uni-heidelberg.de)

WS 12/13



# Parallele Rechnerarchitektur II

- Multiprozessor Architekturen
- Nachrichtenaustausch
- Netzwerktopologien
- Architekturbeispiele
- Routing
- TOP 500
- TOP2 Architekturen



# Einteilung von MIMD-Architekturen

- physische Speicheranordnung

- ▶ gemeinsamer Speicher
- ▶ verteilter Speicher

- Adressraum

- ▶ global
- ▶ lokal

- Programmiermodell

- ▶ gemeinsamer Adressraum
- ▶ Nachrichtenaustausch

- Kommunikationsstruktur

- ▶ Speicherkopplung
- ▶ Nachrichtenkopplung

- Synchronisation

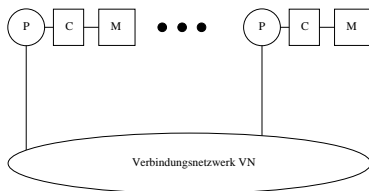
- ▶ Semaphore
- ▶ Barriers

- Latenzbehandlung

- ▶ Latenz verstecken
- ▶ Latenz minimieren



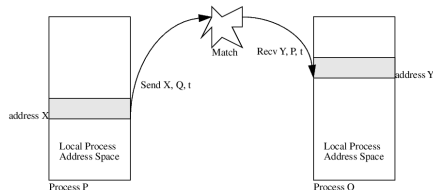
# Distributed Memory: MP



- Multiprozessoren haben einen *lokalen Adressraum*: Jeder Prozessor kann nur auf seinen Speicher zugreifen.
- Interaktion mit anderen Prozessoren ausschließlich über das Senden von Nachrichten.
- Prozessoren, Speicher und Cache sind Standardkomponenten: Volle Ausnutzung des Preisvorteils durch hohe Stückzahlen.
- Verbindungsnetzwerk von Fast Ethernet bis Infiniband.
- Ansatz mit der höchsten Skalierbarkeit: IBM BlueGene > 100 K Prozessoren



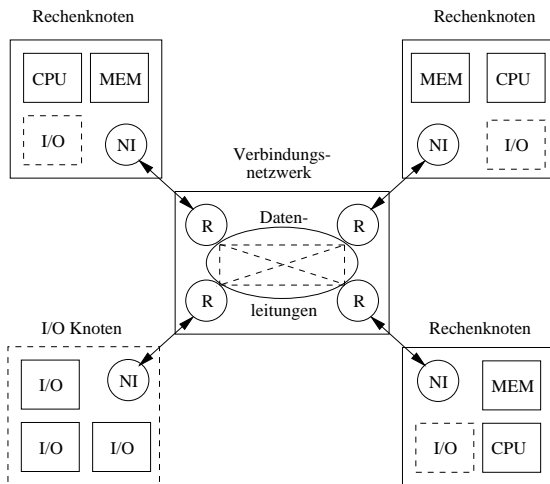
# Distributed Memory: Message Passing



- Prozesse kommunizieren Daten zwischen verteilten Adreßräumen
- expliziter Nachrichtenaustausch notwendig
- Sende-/Empfangsoperationen



# Eine generische Parallelrechnerarchitektur



Generischer Aufbau eines skalierbaren Parallelrechners mit verteiltem Speicher



# Skalierbarkeit: Parameter

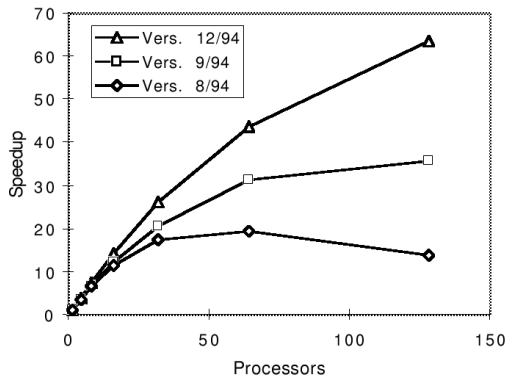
Parameter, welche die Skalierbarkeit eines Systems bestimmen:

- Bandbreite [MB/s]
- Latenzzeit [ $\mu$ s]
- Kosten [\$]
- Physische Größe [ $m^2, m^3$ ]
- Energieverbrauch [W]
- Fault tolerance / recovery abilities

Eine skalierbare Architektur sollte harte Grenzen vermeiden!



# Einfluß von Parallelen Architekturen?



from Culler, Singh, Gupta: Parallel Computer Architecture

Eine skalierbare Architektur ist Voraussetzung für skalierbares Rechnen



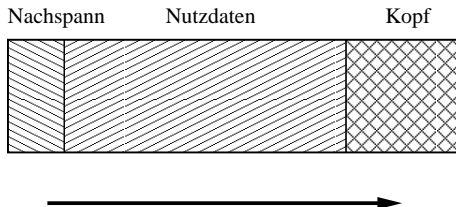


# Nachrichtenaustausch

Speicherblock (variabler Länge) soll von einem Speicher zum anderen kopiert werden

Genauer: Vom Adressraum eines Prozesses in den eines anderen (auf einem anderen Prozessor)

Das Verbindungsnetzwerk ist paketorientiert. Jede Nachricht wird in Pakete fester Länge zerlegt (z. B. 32 Byte bis 4 KByte)



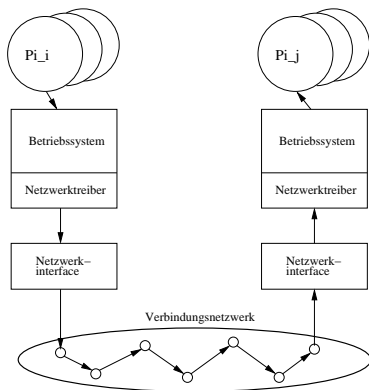
Kopf: Zielprozessor, Nachspann: Prüfsumme

Kommunikationsprotokoll: Bestätigung ob Paket (richtig) angekommen, Flusskontrolle



# Nachrichtenaustausch

Schichtenmodell (Hierarchie von Protokollen):



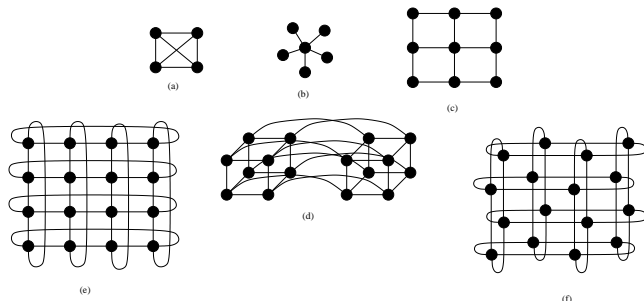
Modell der Übertragungszeit:

$$t_{mess}(n) = t_s + n * t_b.$$

$t_s$ : setup-Zeit (latency),  $t_b$ : Zeit pro Byte,  $1/t_b$ : Bandbreite, abh. von Protokoll



# Netzwerktopologien I



- *Hypercube*: der Dimension  $d$  hat  $2^d$  Prozessoren. Prozessor  $p$  ist mit  $q$  verbunden wenn sich deren Binärdarstellungen *in genau einem Bit* unterscheiden.
- **Netzwerkknoten**: Früher (vor 1990) war das der Prozessor selbst, heute sind es dedizierte Kommunikationsprozessoren



# Netzwerktopologien II

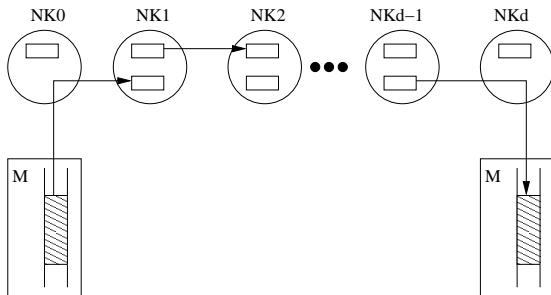
Kennzahlen:

Netzwerktopologie	Knoten- grad K	Leitungs- anzahl L	Durch- messer D	Bisektions- bandbreite B	Sym- metrie
Volle Konnektivität	$N - 1$	$N(N - 1)/2$	1	$(N/2)^2$	Ja
Stern	$N - 1$	$N - 1$	2	$\lfloor N/2 \rfloor$	nein
2D-Gitter	4	$2N - 2\sqrt{N}$	$2(\sqrt{N} - 1)$	$\sqrt{N}$	nein
3D-Torus	6	$3N$	$3\lfloor \sqrt{N}/2 \rfloor$	$2\sqrt{N}$	ja
Hypercube	$\log_2 N$	$nN \log_2 N$	$n$	$N/2$	ja
k-ärer n-cube ( $N = k^n$ )	$3N$	$n\lfloor k/2 \rfloor$	$nN$	$2k^{n-1}$	ja



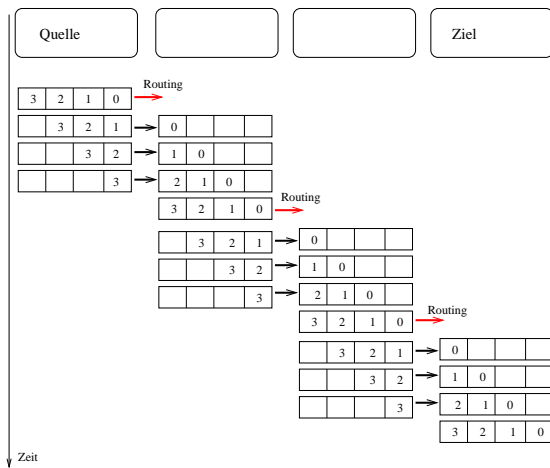
# Store & Forward Routing

*Store-and-forward routing:* Nachricht der Länge  $n$  wird in Pakete der Länge  $N$  zerlegt. Pipelining auf Paketebene: Paket wird aber vollständig im NK gespeichert



# Store & Forward Routing

Übertragung eines Paketes:



# Store & Forward Routing

Laufzeit:

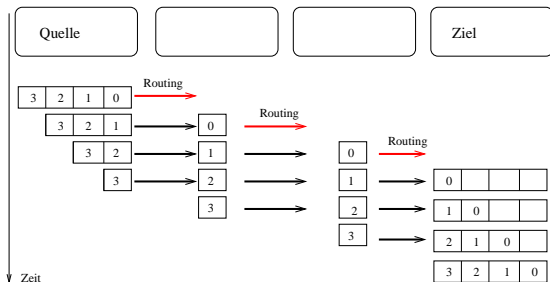
$$\begin{aligned}t_{SF}(n, N, d) &= t_s + d(t_h + Nt_b) + \left(\frac{n}{N} - 1\right) (t_h + Nt_b) \\ &= t_s + t_h \left(d + \frac{n}{N} - 1\right) + t_b (n + N(d - 1)).\end{aligned}$$

- $t_s$ : Zeit, die auf Quell- und Zielrechner vergeht bis das Netzwerk mit der Nachrichtenübertragung beauftragt wird, bzw. bis der empfangende Prozess benachrichtigt wird. Dies ist der Softwareanteil des Protokolls.
- $t_h$ : Zeit die benötigt wird um das erste Byte einer Nachricht von einem Netzwerkknoten zum anderen zu übertragen (*engl.* node latency, hop-time).
- $t_b$ : Zeit für die Übertragung eines Byte von Netzwerkknoten zu Netzwerkknoten.
- $d$ : Hops bis zum Ziel.



# Cut-Through Routing

*Cut-through routing* oder *wormhole routing*: Pakete werden nicht zwischengespeichert, jedes Wort (sog. *flit*) wird sofort an nächsten Netzwerkknoten weitergeleitet  
Übertragung eines Paketes:



Laufzeit:

$$t_{CT}(n, N, d) = t_s + t_h d + t_b n$$

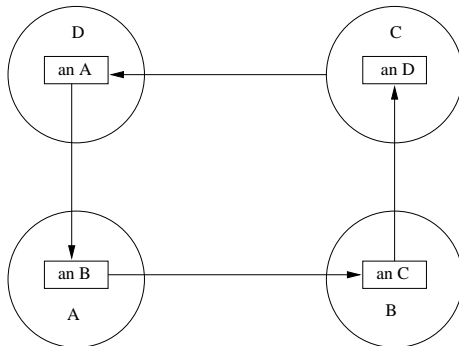
Zeit für kurze Nachricht ( $n = N$ ):  $t_{CT} = t_s + dt_h + Nt_b$ . Wegen  $dt_h \ll t_s$  (Hardware!) quasi entfernungsunabhängig





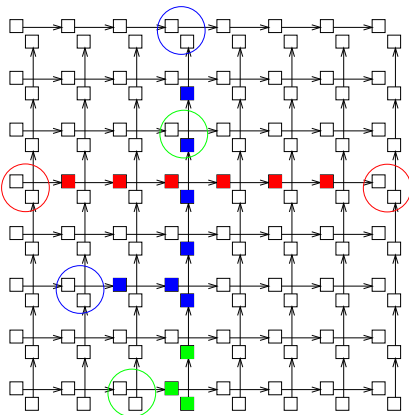
# Deadlock

In paketvermittelnden Netzwerken besteht die Gefahr des *store-and-forward deadlock*:



# Deadlock

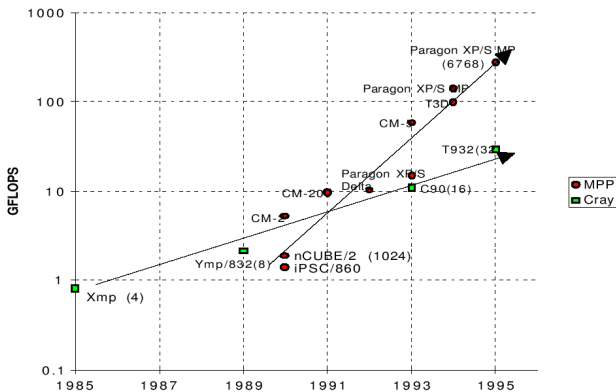
Zusammen mit cut-through routing:



Verklemmungsfreies „dimension routing“. Beispiel 2D-Gitter: Zerlege Netzwerk in  $+x$ ,  $-x$ ,  $+y$  und  $-y$  Netzwerke mit jeweils eigenen Puffern. Nachricht läuft erst in Zeile, dann in Spalte.



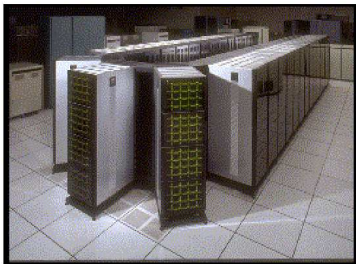
# Multi-Processor Performance



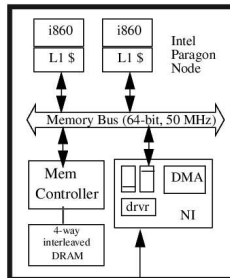
from Culler, Singh, Gupta: Parallel Computer Architecture



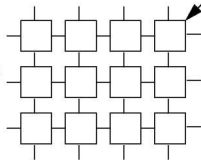
# Message Passing Architectures I



Source: <http://www.cs.sandia.gov/gif/paragon.gif>  
courtesy of Sandia National Laboratory  
1824 nodes configured as a 16  
high by 114 wide array



2D grid network  
with processing node  
attached to every switch



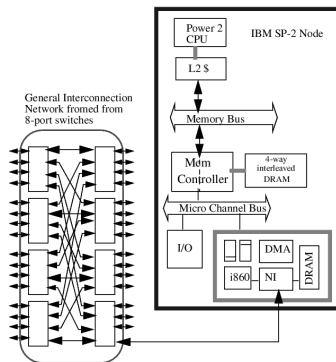
8 bits  
175 MHz  
bidirectional

## Intel Paragon:

- erste Machine mit parallelem Unix
- Prozeßmigration, Gangscheduling



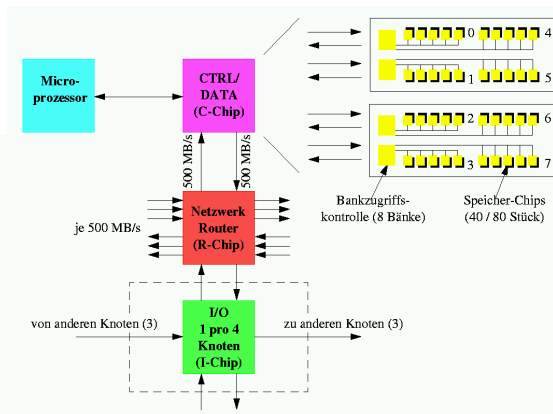
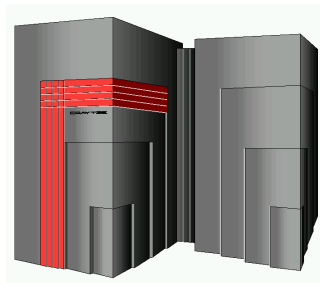
# Message Passing Architectures II



## IBM SP2:

- Rechenknoten sind RS 6000 workstations
- Switching Netzwerk

# Message Passing Architectures III



Cray T3E:

- hohe Packungsdichte
- ein systemweiter Taktgeber
- virtual shared memory



## Top500 Benchmark:

- LINPACK Benchmark wird zur Evaluation der Systeme verwendet
- Benchmarkleistung reflektiert nicht die Gesamtleistung des Systems
- Benchmark zeigt Performanz bei der Lösung von dichtbesetzten linearen GLS
- Sehr reguläres Problem: erzielte Leistung ist sehr hoch (nahe peak performance)



# Top 10 of Top500

Rank	Site	Computer/Year Vendor	Cores	$R_{max}$	$R_{peak}$	Power
1	DOE/NNSA/LLNL United States	<b>Sequoia</b> - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom / 2011 IBM	1572864	16324.75	20132.66	7890.0
2	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu Interconnect / 2011 Fujitsu	705024	10510.00	11280.38	12659.9
3	DOE/SC/Argonne National Laboratory United States	<b>Mira</b> - BlueGene/Q, Power BQC 16C 1.60GHz, Custom / 2012 IBM	786432	8162.38	10066.33	3945.0
4	Leibniz Rechenzentrum Germany	<b>SuperMUC</b> - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband FDR / 2012 IBM	147456	2897.00	3185.05	3422.7
5	National Supercomputing Center in Tianjin China	<b>Tianhe-1A</b> - NUDT YH MPP, Xeon X5670 8C 2.93 GHz, NVIDIA 2050 / 2010 NUDT	186368	2566.00	4701.00	4040.0
6	DOE/SC/Oak Ridge National Laboratory United States	<b>Jaguar</b> - Cray XK6, Opteron 6274 16C 2.200GHz, Cray Gemini Interconnect, NVIDIA 2090 / 2009 Cray Inc.	298592	1941.00	2627.61	5142.0
7	CINECA Italy	<b>Fermi</b> - BlueGene/Q, Power BQC 16C 1.60GHz, Custom / 2012 IBM	163840	1725.49	2097.15	821.9
8	Forschungszentrum Juelich (FZJ) Germany	<b>JuQUEEN</b> - BlueGene/Q, Power BQC 16C 1.60GHz, Custom / 2012 IBM	131072	1380.39	1677.72	657.5
9	CEA/TGCC-GENCI France	<b>Curie thin nodes</b> - Bullx B510, Xeon E5-2680 8C 2.700GHz, Infiniband QDR / 2012 Bull	77184	1359.00	1667.17	2251.0
10	National Supercomputing Centre in Shenzhen (NSCS) China	<b>Nebulae</b> - Dawning TC3600 Blade System, Xeon X5650 6 C 2.66GHz, Infiniband QDR, NVIDIA 2050 / 2010 Dawning	120640	1271.00	2984.30	2580.0



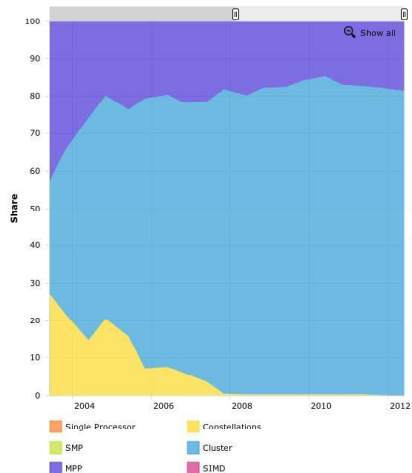
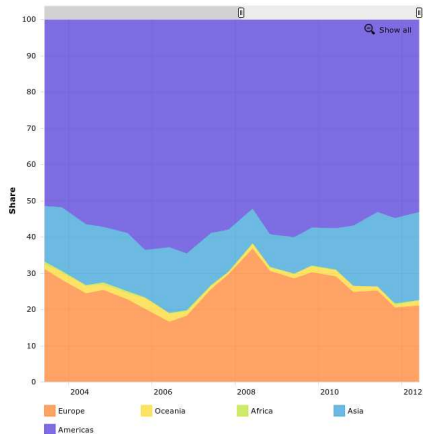


# Top500 key facts

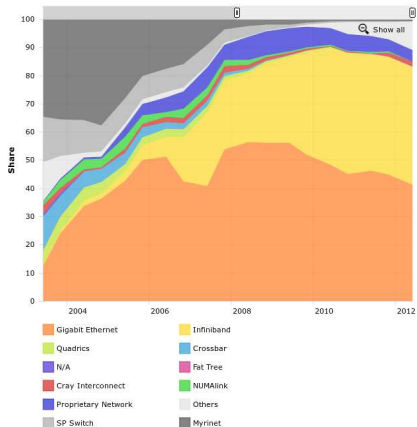
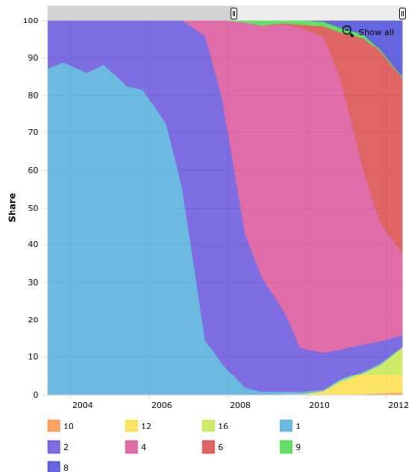
- Eintrittsbarriere ist Leistung von 60.8 TeraFlop/s
- Mittlerer Energieverbrauch der Top10 ist 4.09 MW: 0.8-2 GFlops/W
- 40 Systeme verbrauchen mehr als 1 MW
- Akkumulierte Leistung ist 123.4 PFlops/s (74.2 PFlop/s)
- Top100 Mindestleistung ist 172.6 TFlop/s (115.9 TFlop/s)
- 20 Petaflops-Systeme
- TOP 100 Mindestleistung 12.97 TFlop/s (9.29 TFlop/s)
- Prozessortyp: Intel SandyBridge, AMD Opteron, IBM Power 68
- 74.8 % der Systeme verwenden Prozessoren mit 6 oder mehr Kernen
- Infiniband (208) and Gigabit Ethernet (207) Netzwerke dominieren
- Architektur: 80% Cluster, 20% MPP, 0% SIMD/SMP



# Top500 Continent + Architecture Type

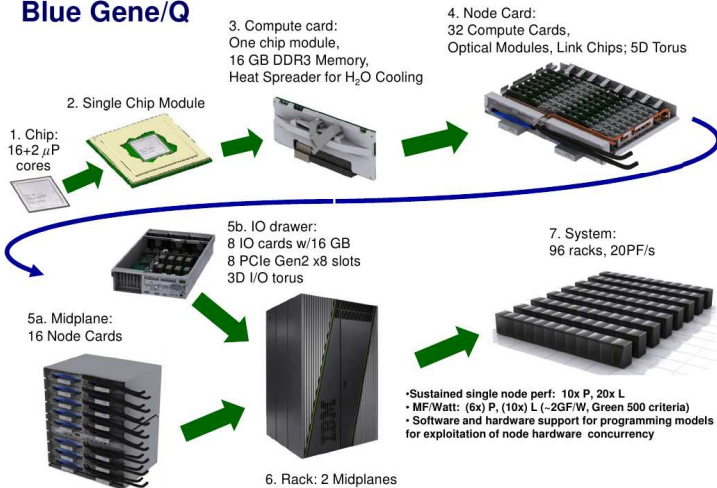


# Top500 CoresPerSocket + Interconnect



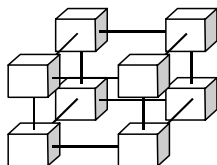
# IBM Blue Gene/Q Architecture

## Blue Gene/Q



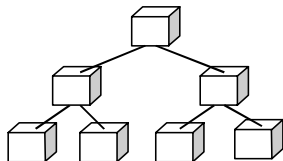
# IBM Blue Gene/Q Networks

98304 Knoten verbunden über drei integrierte Netzwerke



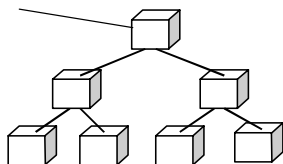
## 3 Dimensional Torus

- Virtual cut-through hardware routing to maximize efficiency
- 2.8 Gb/s on all 12 node links (total of 4.2 GB/s per node)
- Communication backbone
- 134 TB/s total torus interconnect bandwidth



## Global Tree

- One-to-all or all-all broadcast functionality
- Arithmetic operations implemented in tree
- ~1.4 GB/s of bandwidth from any node to all other nodes
- Latency of tree traversal less than 1usec



## Ethernet

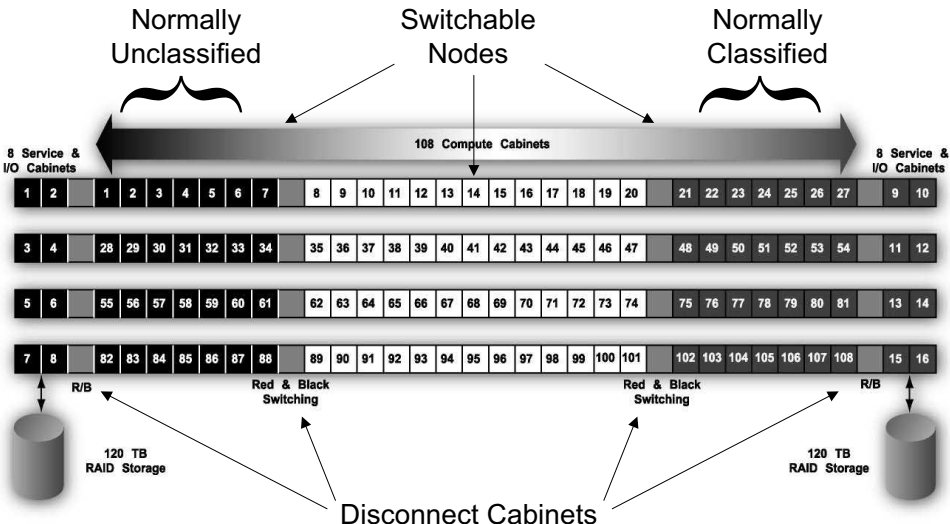
- Incorporated into every node ASIC
- Disk I/O
- Host control, booting and diagnostics



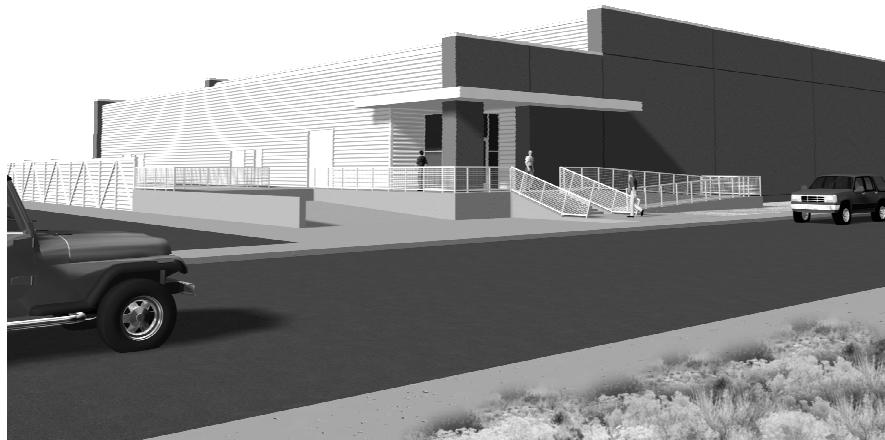
# Cray RedStorm



# Cray RedStorm Configuration

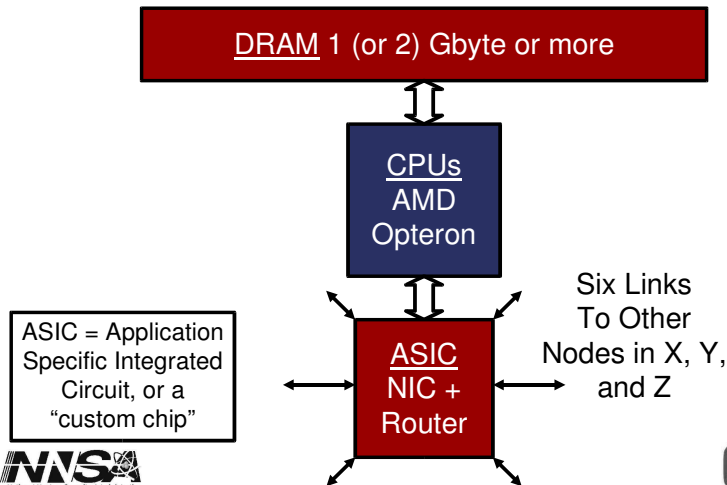


# Cray RedStorm Building

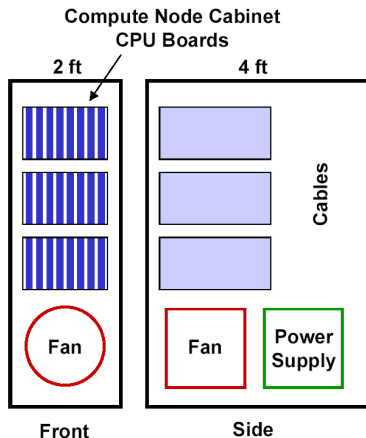




# Cray RedStorm Compute Node



# Cray RedStorm Cabinet



- **Compute Node Cabinet**
  - ♦ 3 Card Cages per Cabinet
  - ♦ 8 Boards per Card Cage
  - ♦ 4 Processors per Board
  - ♦ 4 NIC/Router Chips per Board
  - ♦ N + 1 Power Supplies
  - ♦ Passive Backplane
- **Service and I/O Node Cabinet**
  - ♦ 2 Card Cages per Cabinet
  - ♦ 8 Boards per Card Cage
  - ♦ 2 Processors per Board
  - ♦ 4 NIC/Router Chips per Board
  - ♦ Dual PCI-X for each processor
  - ♦ N + 1 Power Supplies
  - ♦ Passive Backplane



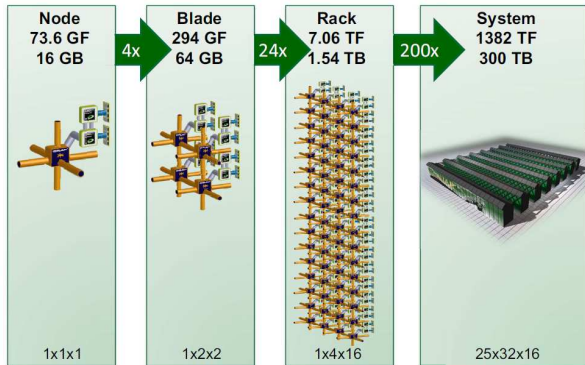
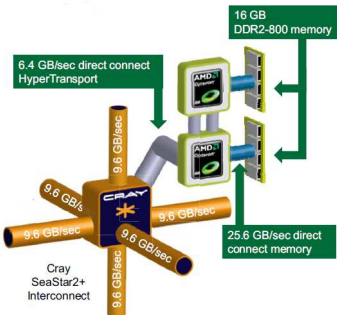
# Blue Gene L vs Red Storm

BGL 360 TF version, Red Storm 100 TF version

	Blue Gene L	Red Storm	
Node speed	5.6 GF	5.6 GF	(1x)
Node memory	.25 - .5 GB	2 (1-8 GB)	(4x)
Network latency	7 us	2 us	(2/7x)
Network link bw	0.28 GB/s	6.0 GB/s	(22x)
BW Bytes/Flops	0.05	1.1	(22x)
Bi-Section B/F	0.0016	0.038	(24x)
#nodes/problem	40,000	10,000	(1/4x)



# Cray XT-5 Jaguar Architecture



# Cray XT-5 Jaguar IO-Configuration

