Rechnerstrukturen

Vorlesung im Sommersemester 2007

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik

- **Kapitel 3: Multiprozessoren – Parallelismus auf Prozess/Thread-Ebene**

3.5: Multiprozessoren mit gemeinsamem Speicher

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Cache-Kohärenz
    - sichert, dass mehrere Prozessoren eine kohärente Sicht auf den Speicher haben
  - Wichtige Frage:
    - Wann muss ein Prozessor den Wert sehen, den ein anderer Prozessor aktualisiert hat?
    - Oder:
      - In welcher Reihenfolge muss ein Prozessor die Schreiboperationen eines anderen Prozessors beobachten?
    - Oder
      - Welche Bedingungen zwischen Lese- und Schreiboperationen auf verschiedene Speicherstellen durch verschiedene Prozessoren müssen gelten?

- Speicherkonsistenzmodelle
  - Problem: Was kann passieren?

```
P1:     A=0;                          P2:     B=0;
        …                                     …
        A=1;                                  B=1;
L1:     if (B==0)                     L2:     if (A==0)
                …führe Aktion a2 aus                  …führe Aktion a1 aus
```

Mögliche Fälle:

a1 wird ausgeführt und a2 nicht
a2 wird ausgeführt und a1 nicht
a1 und a2 werden beide nicht ausgeführt
a1 und a2 werden beide ausgeführt

Ursache für dieses Verhalten:

- Verzögerungen der Schreiboperationen im Schreibpuffer
- Verzögerungen im Netzwerk

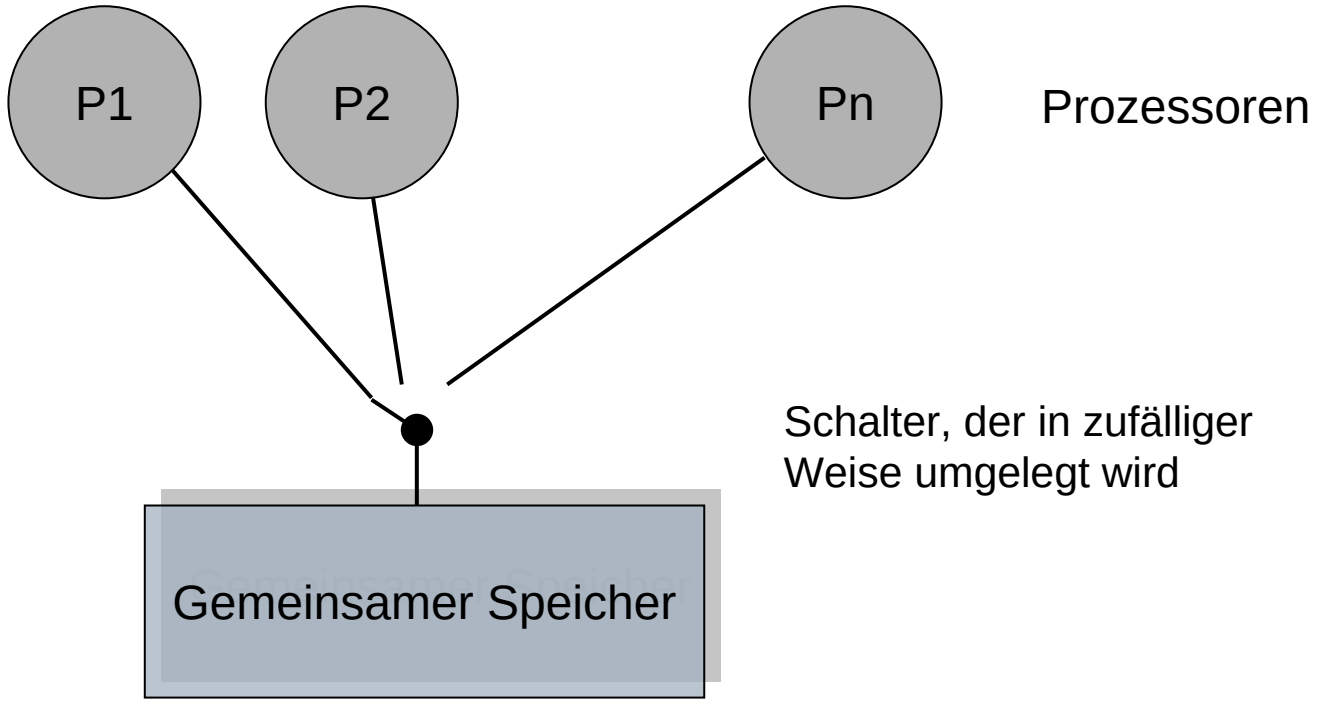
Soll dieses Verhalten erlaubt sein, und wenn ja, unter welchen Bedingungen?

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Spezifizieren die Reihenfolge, in der Speicherzugriffe eines Prozesses von anderen Prozessen gesehen werden
  - Sequentielle Konsistenz
    - Ein Multiprozessorsystem heißt sequentiell konsistent, wenn das Ergebnis einer beliebigen Berechnung dasselbe ist, als wenn die Operationen aller Prozessoren auf einem Einprozessorsystem in einer sequentiellen Ordnung ausgeführt würden. Dabei ist die Ordnung der Operationen der Prozessoren die des jeweiligen Programms.
    - Alle Lese- und Schreibzugriffe werden in einer beliebigen sequentiellen Reihenfolge, die jedoch mit den jeweiligen Programmordnungen konform ist, am Speicher wirksam.
    - Entspricht einer überlappenden sequentiellen Ausführung sequentieller Operationsfolgen anstelle einer parallelen Ausführung

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Sequentielle Konsistenz
    - Veranschaulichung der sequentiellen Konsistenz



Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Sequentielle Konsistenz
    - Programmierer geht von sequentieller Konsistenz aus
    - Führt aber zu sehr starken Einbußen bzgl. Implementierung und damit der Leistung
      - Verbietet vorgezogene Ladeoperationen, nichtblockierende Caches
  - Abgeschwächte Konsistenzmodelle
    - Konsistenz nur zum Zeitpunkt einer Synchronisationsoperation
      - Lese- und Schreiboperationen der parallel arbeitenden Prozessoren auf den gemeinsamen Speicher zwischen den Synchronisationszeitpunkten können in beliebiger geschehen.

- Speicherkonsistenzmodelle
  - Definitionen
    - Ein Lesezugriff durch Prozessor Pi heißt zu einem bestimmten Zeitpunkt bezüglich Pk ausgeführt, wenn ein Schreibzugriff durch Prozessor Pk den Wert, den Pi durch den Lesezugriff auf dieselbe Adresse erhält, nicht mehr beeinflussen kann
    - Ein Schreibzugriff durch Pi heißt zu einem bestimmten Zeitpunkt bezüglich Pk ausgeführt, wenn ein Lesezugriff durch Pk auf dieselbe Adresse den Wert liefert, der von Pi geschrieben worden ist
    - Ein Zugriff gilt als ausgeführt, wenn er bezüglich aller Prozessoren im System ausgeführt ist
    - Ein Lesezugriff heißt global ausgeführt, wenn sowohl er als auch der Schreibzugriff, der den gelesenen Wert erzeugt, ausgeführt worden ist

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Definitionen
    - Unterschied zwischen ausgeführten und global ausgeführten Lesezugriff nur in Systemen, in denen der Schreibzugriff kein atomarer Vorgang ist, d.h. wenn der geschriebene Wert für alle Prozessoren des Systems nicht gleich lesbar ist
  - Sequentielle Konsistenz:
    - Hinreichende Bedingung:
      - Bevor ein Lese- oder Schreibzugriff bezüglich eines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Lesezugriffe global ausgeführt und alle vorhergehenden Schreibzugriffe ausgeführt sein
    - Reihenfolge der Operationen auf einem Prozessor wird beibehalten, und für alle anderen Prozessoren ist dieselbe Reihenfolge sichtbar.
    - Jeder Lese- und Schreibzugriff muss allen anderen Prozessoren vor einem nachfolgenden Lese- oder Aschreibzugriff bekannt gemacht werden
    - Wenig Spielraum für Optimierungen der Speicherzugriffe
      - Z.B. Puffern der Schreibzugriffe

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Prozessorkonsistenz
    - Definition (nach Goodman, 1989)
      - Ein Prozessor ist prozessorkonsistent, wenn das Ergebnis irgendeiner Ausführung dasselbe ist, als wenn die Operation eines jeden Prozessors in der sequentiellen Reihenfolge, die durch sein Programm bestimmt wird, erscheinen
    - Unterschied zur sequentiellen Konsistenz
      - Bei der Prozessorkonsistenz muss es nicht mehr für alle Prozessoren eine einheitliche Reihenfolge der Speicherzugriffe geben
      - Schreibzugriffe zweier Prozessoren können von einem dritten Prozessor in einer anderen Reihenfolge gesehen werden, als von den beiden schreibenden Prozessoren
      - Die Schreibzugriffe eines Prozessors werden jedoch von allen Prozessoren in der in seinem Programm angegebenen Reihenfolge gesehen

- Speicherkonsistenzmodelle
  - Prozessorkonsistenz
    - Definition (nach Gharachorloo, 1990)
      - Bedingung der globalen Ausführung der Lesezugriffe wird fallen gelassen
      - Es gilt:
        - » Bevor ein Lesezugriff bezüglich eines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Lesezugriffe ausgeführt worden sein
        - » Bevor ein Schreibzugriff irgendeines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Schreibzugriffe ausgeführt worden sein

- Speicherkonsistenzmodelle
  - Prozessorkonsistenz
    - Definition (nach Gharachorloo, 1990)
      - Konsequenz
        - » Prozessorkonsistenz führt nicht mehr unbedingt zur korrekten Sequentialisierung der Speicherzugriffe, da das Lesen schon erlaubt ist, bevor die vorhergehenden Schreibzugriffe alle ausgeführt worden sind.
        - » Lesezugriff kann zwar von den anderen Prozessoren nicht beobachtet werden, der Prozessor selbst sieht jedoch eine Reihenfolge der Speicheroperationen, die nicht seiner Programmordnung entspricht
        - » Schreibender Prozessor muss nicht auf die Ausführung des Schreibzugriffs bezüglich aller Prozessoren warten, bevor er eine weitere Schreiboperation veranlassen kann
        - » Puffern von Schreibzugriffen ist wieder erlaubt

- Speicherkonsistenzmodelle
  - Schwache Konsistenz
    - Bisherige Konsistenzmodelle lassen Synchronisation paralleler Threads außer Acht
    - Konkurrierende Zugriffe auf gemeinsame Daten werden durch geeignete Synchronisationen geschützt

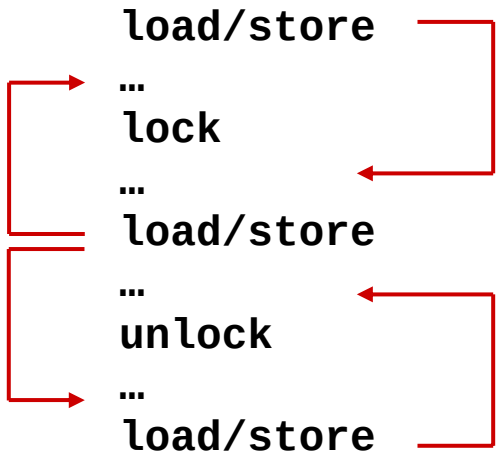
```
mutex m;
…
lock(m)
y=0;
x=0;
unlock(m)
…
P1:     lock(m)
        A=1;
        if (B==0)
                …führe Aktion a2 aus
        unlock(m)

P2:     lock(m)
        B=1;
        if (A==0)
                …führe Aktion a1 aus
        unlock(m)
```

- Speicherkonsistenzmodelle
  - Schwache Konsistenz (weak consistency)
    - Idee
      - Die Konsistenz des Speicherzugriffs wird nicht mehr zu allen Zeiten gewährleistet, sondern zu bestimmten, vom Programmierer in das Programm eingesetzten Synchronisationspunkten
      - Einführung von kritischen Bereichen
        - » Innerhalb dieser Bereich wird die Inkonsistenz der gemeinsamen Daten zugelassen
        - » Voraussetzung: konkurrierende Lese-/Schreibzugriffe sind durch den kritischen Bereich unterbunden

- Speicherkonsistenzmodelle
  - Schwache Konsistenz (weak consistency)
    - Bedingungen
      - Bevor ein Schreib- oder Lesezugriff bezüglich irgendeines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Synchronisationspunkte erreicht worden sein
      - Bevor eine Synchronisation bezüglich irgendeines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Schreib- oder Lesezugriffe ausgeführt worden sein.
      - Synchronisationspunkte müssen sequentiell konsistent sein
        - » *bevor* und *vorübergehend* beziehen sich auf die Programmordnung

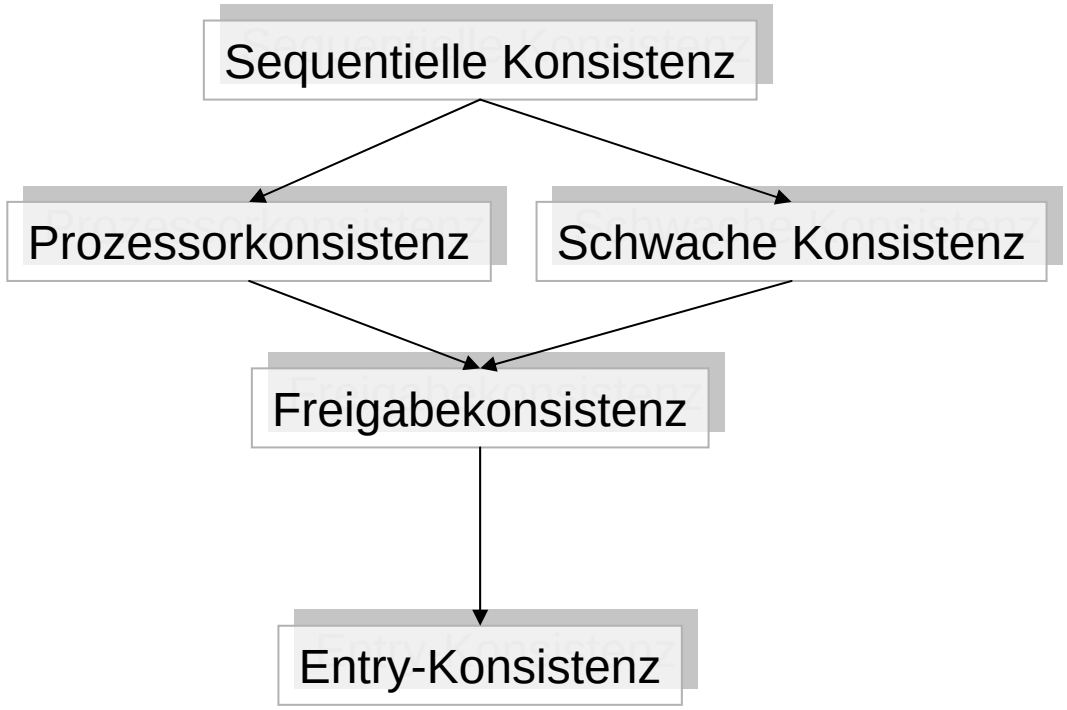
- Speicherkonsistenzmodelle
  - Schwache Konsistenz (weak consistency)
    - Auswirkung
      - Synchronisationsbefehle stellen Hürden dar, die von keinem Lese- oder Schreibzugriff übersprungen werden



Rote Pfeile: verboten

- Speicherkonsistenzmodelle
  - Schwache Konsistenz (weak consistency)
    - Auswirkung
      - Voraussetzung für die Implementierung der schwachen Konsistenz ist die hardware- und softwaremäßige Unterscheidung der Synchronisationsbefehle von den Lade- und Speicherbefehlen und eine sequentiell konsistente Implementierung der Synchronisationsbefehle
      - Das Puffern von Schreibzugriffen ist erlaubt, das von Synchronisationsbefehlen nicht!
      - „Hürdeneigenschaft“ der Synchronisationsbefehle
        - » In heutigen Mikroprozessoren mit Hilfe von Spezialbefehlen implementiert
      - Die Ordnung der Speicherbefehle wird durch die sehr viel losere Ordnung der Synchronisationsbefehle ersetzt

- Speicherkonsistenzmodelle
  - Weitere Konsistenzmodelle
    - Zusammenhang



- Speicherkonsistenzmodelle
  - Weitere Konsistenzmodelle
    - definieren theoretisch weitere Abschwächungen oder
    - leiten sich aus Hardware-Implementierungen spezieller Prozessoren oder Multiprozessorsysteme ab
      - SPARC-Prozessor-spezifisches schwaches Konsistenzmodell TSO (total store order) oder das hiervon abgeleitete PSO (patial total store)

- **Kapitel 3: Multiprozessoren – Parallelismus auf Prozess/Thread-Ebene**

3.6: Multiprozessoren mit verteiltem Speicher

- IBM Blue Gene/L Überblick
  - Massively Parallel Supercomputer
    - Ziel:
      - günstiges Cost/Performance-Verhaltnis für ein breites Spektrum von Anwendungen
      - Günstiges Performance/Power-Verhältnis
    - Grundlegender Ansatz:
      - System-on-Chip Design für den Prozessor
        - » Hohe Integrationsdichte
        - » Low Power
        - » Low Design Cost
      - Hohe Skalierbarkeit der Anwendungen
        - » Hohe Anforderung an die Skalierbarkeit des VErbindungsnetzwerkes

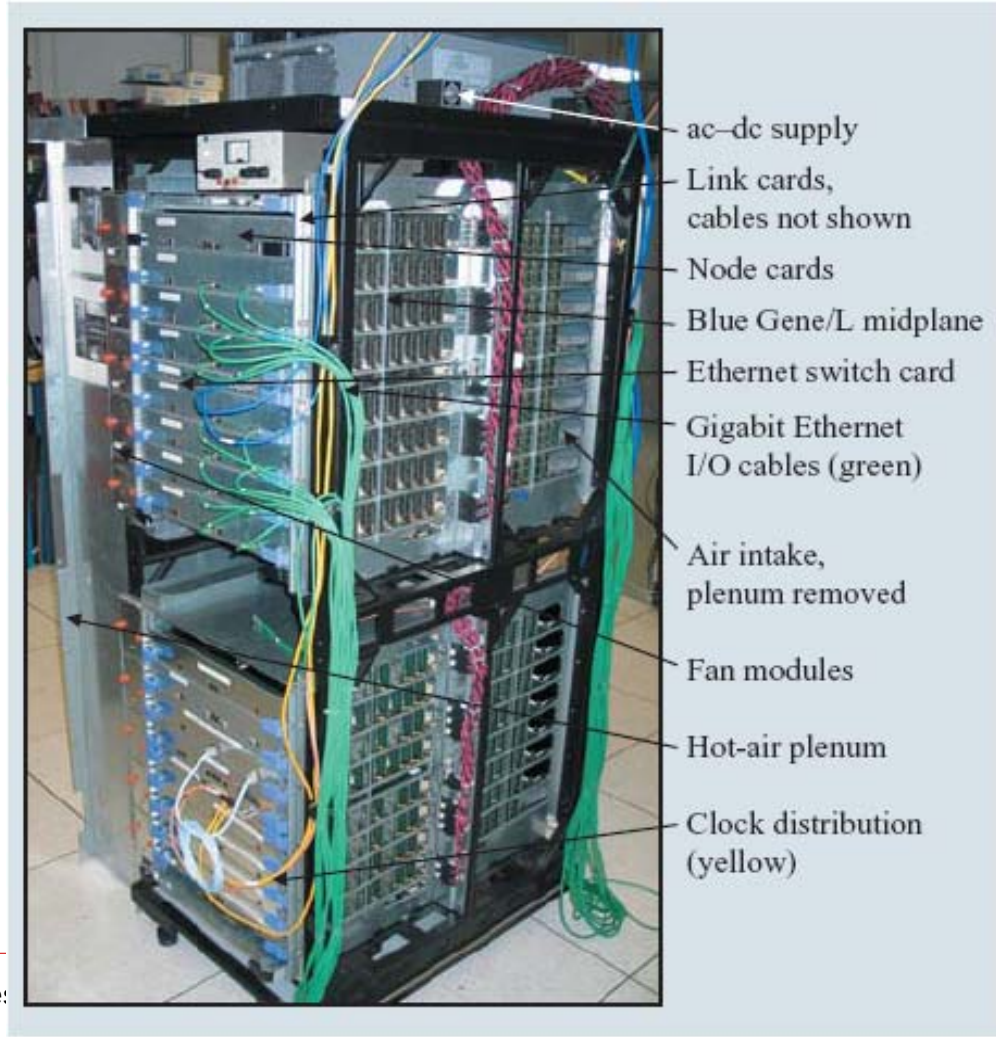
- IBM Blue Gene/L Überblick
  - Massively Parallel Supercomputer
    - Bedeutung Low-Power
      - Für einen Rechner mit einer Leistung im Bereich 380 TFlops mit konventionellen Hochleistungsprozessoren würde der Leistungsverbrauch bei etwa 10 MW – 20 MW liegen, was den Energieverbrauch einer 11000 Einwohner Stadt entspricht.
      - Ein Rack mit 1024 Dual-Prozessor Knoten
        - » Ausmaße: 0.9m x 0.9m x 1,9m
        - » Energieverbrauch: 27,5 kW
    - Bedeutung: Zuverlässigkeit, Verfügbarkeit, Sicherheit (Reliability, Availability, Security, RAS)
    - Bedeutung: Programmierunterstützung
      - Nachrichten-orientiertes Programmiermodell MPI,

- IBM Blue Gene/L Überblick
  - Massively Parallel Supercomputer
    - Anwendungsszenarios:
      - Simulation physikalischer Phänomene
      - Echtzeit-Datenverarbeitung
      - Off-line Datenanalyse
      - Anwendungen in den großen amerikanischen Forschungslabors

- IBM Blue Gene/L Überblick
  - Systemkomponenten
    - 65536 Knoten
      - ASIC: Dual-Processor Chip
      - 18 SDRAM chips
    - Knoten über 5 Netzwerke verbunden
      - Wichtigstes Netzwerk mit höchster Bandbreite
        - » 64 x 32 x 32 3-D Torus

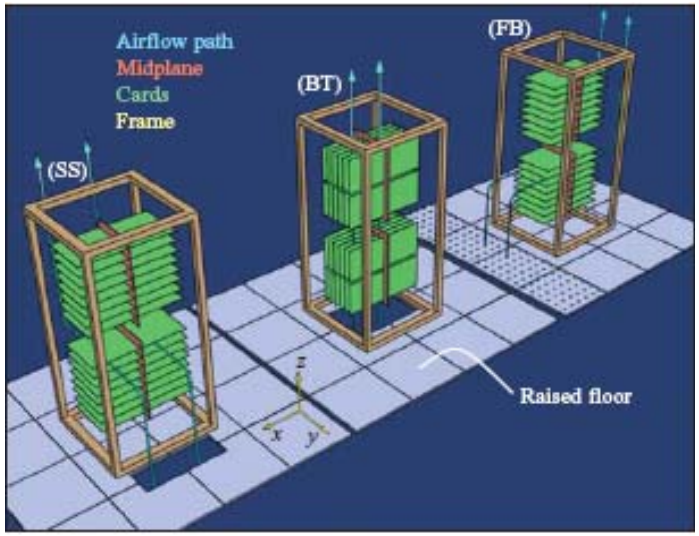
- IBM Blue Gene/L Überblick
  - Systemkomponenten
    - 65536 Knoten bis zu 64 Racks, die auch so organisiert werden können, als wären es verschiedene Systeme, wobei auf jedem ein eigenes Single Software Image läuft
    - Knoten
      - 2 BG/L Compute ASIC (BLC)
        - » Dual Processor SoC ASIC
      - 9 Double data rate synchronous dynamic random access memory chips (DDR SDRAM chips) pro ASIC
    - Knoten über 5 Netzwerke verbunden
      - Wichtigstes Netzwerk mit höchster Bandbreite
        - » 64 x 32 x 32 3-D Torus
        - » Global Collective Network
        - » Global Barrier and Interrupt Network
        - » I/O Network (Gigabit Ethernet)
        - » Service Network

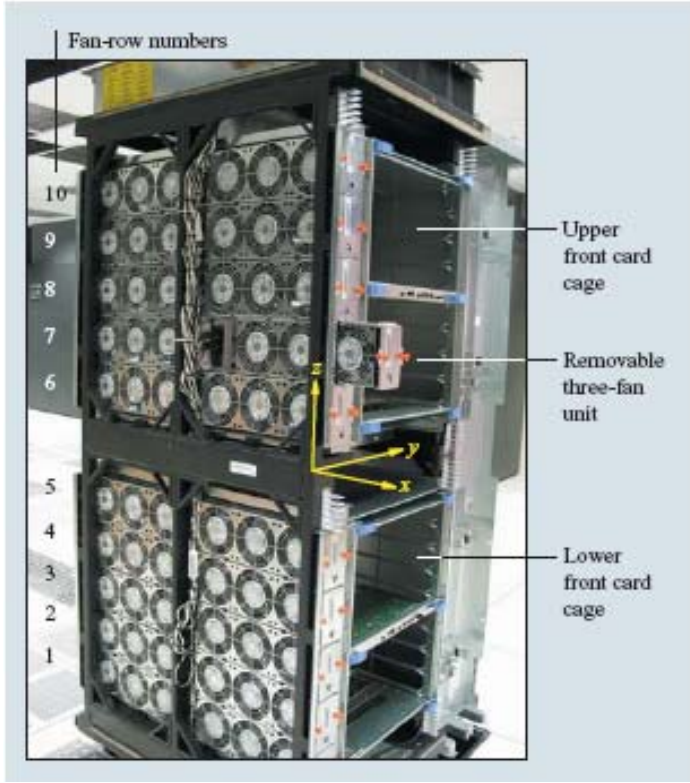
- IBM Blue Gene/L Überblick
  - Systemkomponenten
    - System Rack



Quelle: P. Coteus, et.al.: Packaging the Blue Gene/L supercomputer. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

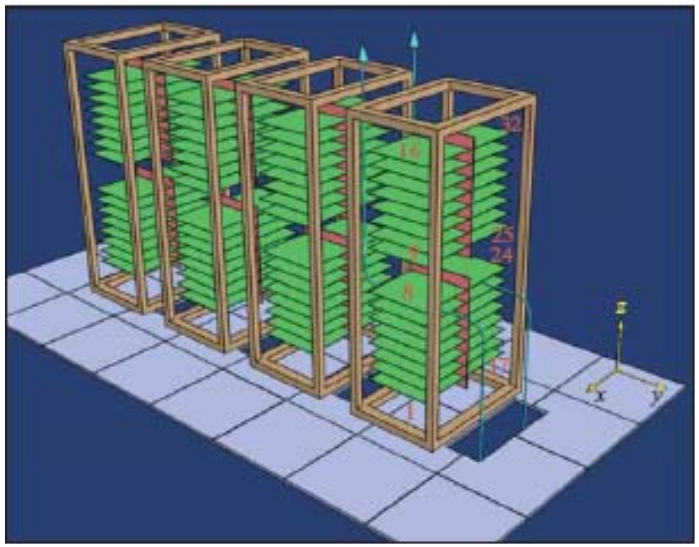
- IBM Blue Gene/L Überblick
  - Systemkomponenten
    - Kühlung

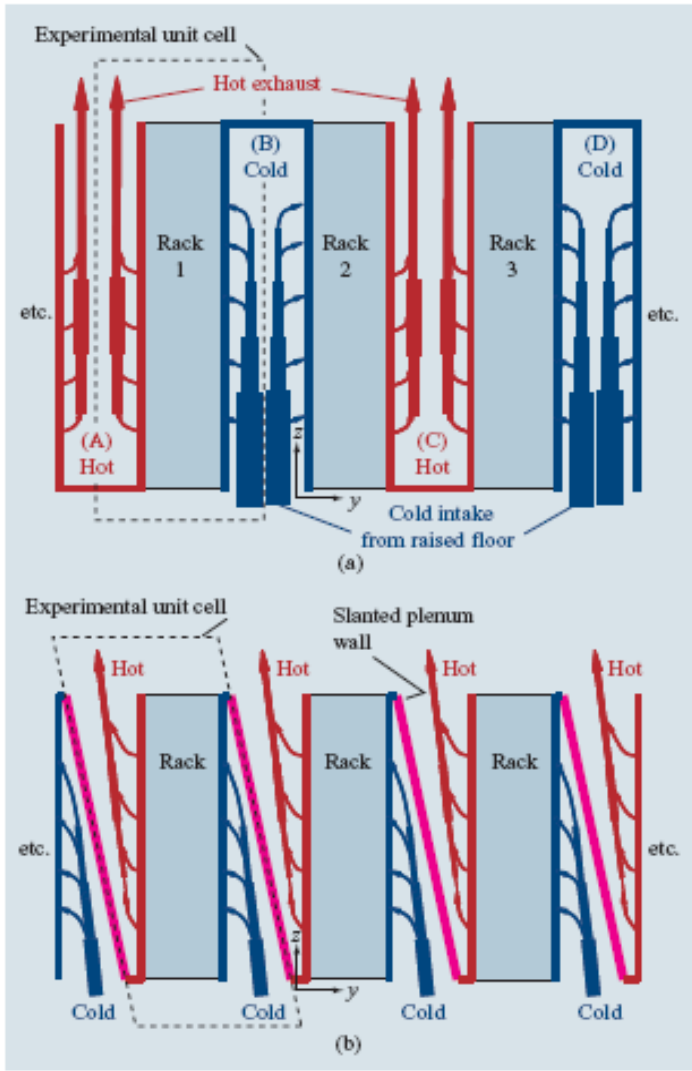




Quelle: P. Coteus, et.al.: Packaging the Blue Gene/L supercomputer. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

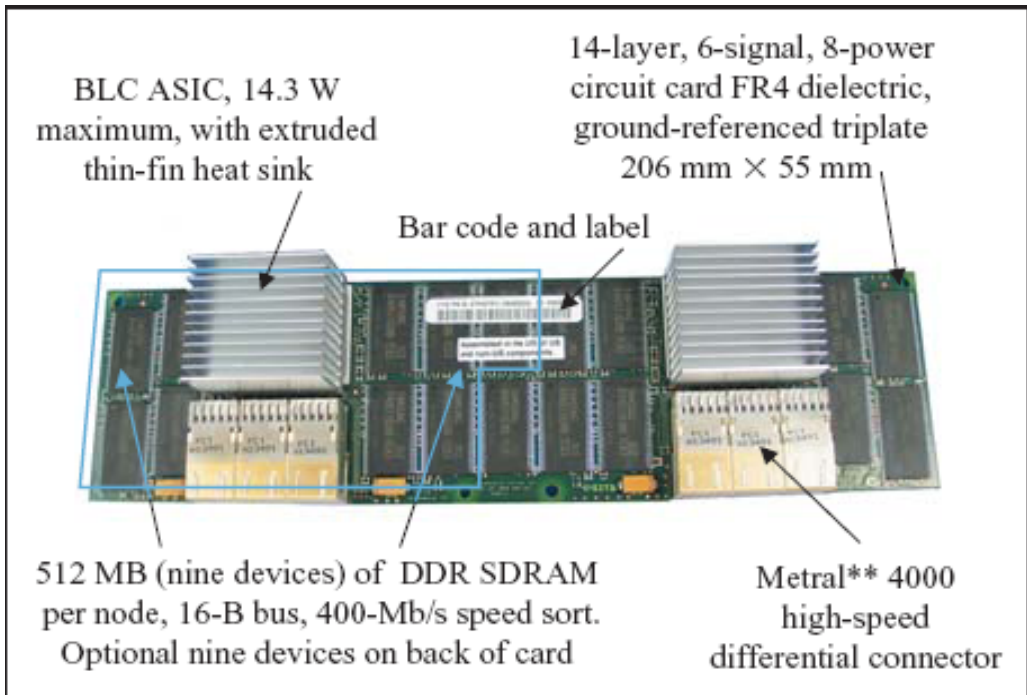
- IBM Blue Gene/L Überblick
  - Systemaufbau





Quelle: P. Coteus, et.al.: Packaging the Blue Gene/L supercomputer. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - Systemkomponenten
    - BG/L Compute card

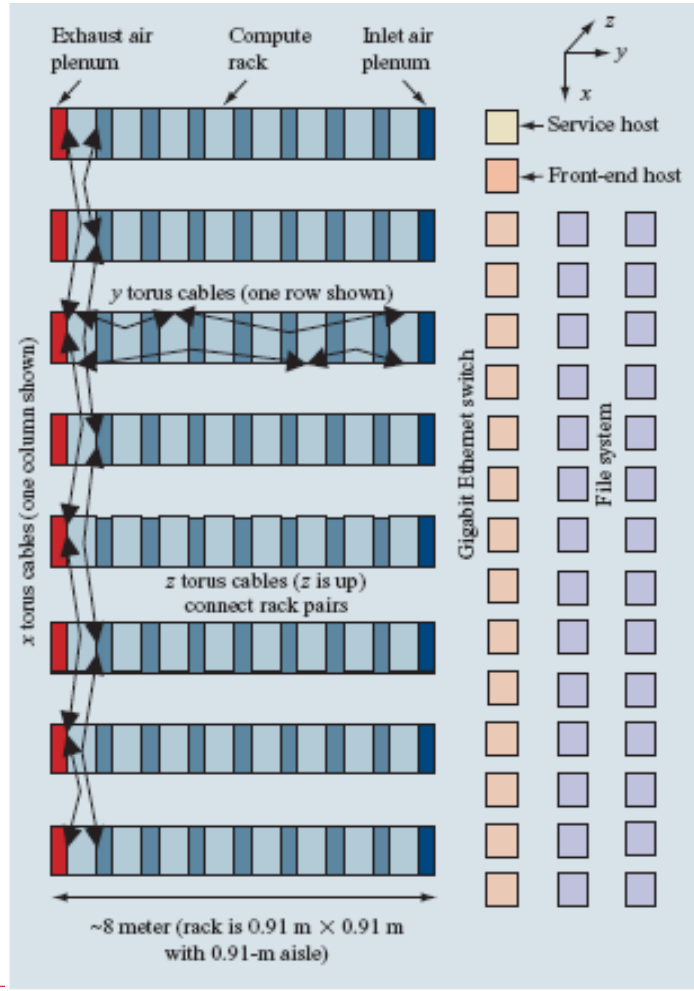


Quelle: P. Coteus, et.al.: Packaging the Blue Gene/L supercomputer. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - Systemkomponenten
    - BG/L Node Card

Quelle: P. Coteus, et.al.: Packaging the Blue Gene/L supercomputer. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

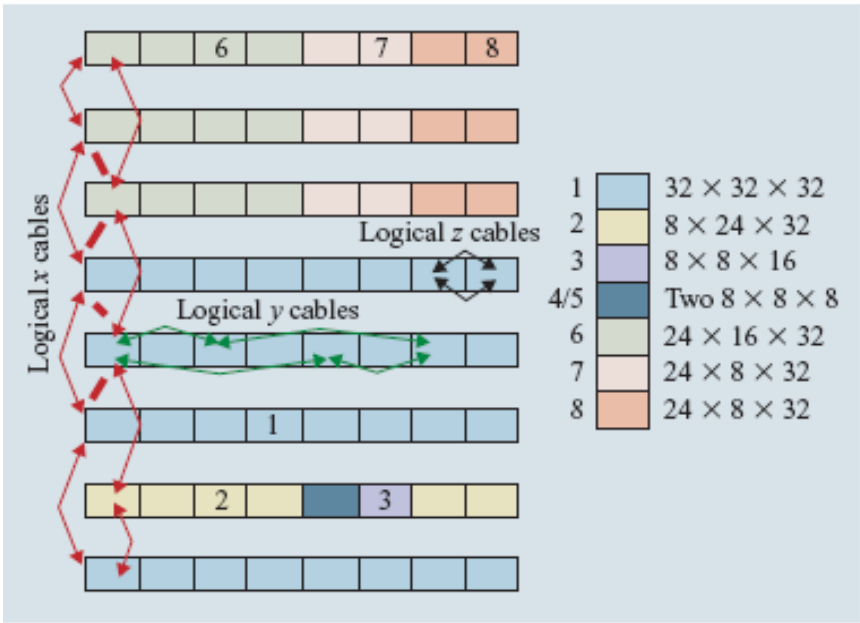
- IBM Blue Gene/L Überblick
  - Ein BG/L System kann für eine Anwendung konfiguriert werden



Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

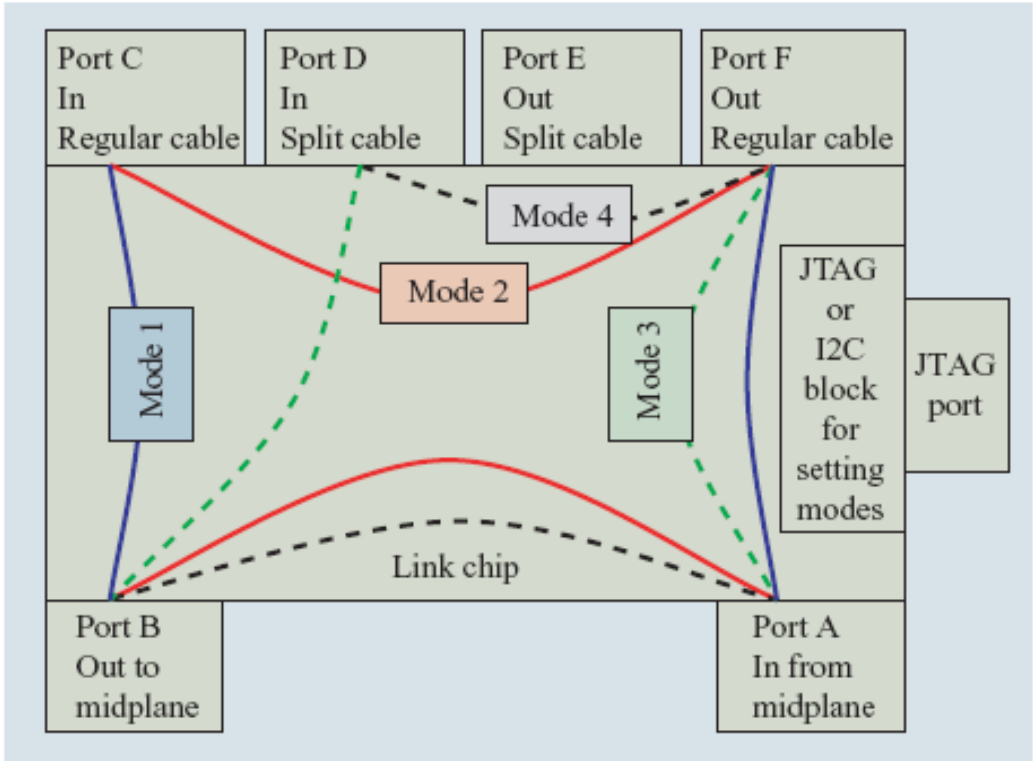
- IBM Blue Gene/L Überblick
  - Systempartitionierung
    - Partitionierung in kleinere Systeme
      - Beispiel System mit 20K Knoten (20 Rack-System)
        - » 4 Reihen mit 4 Compute Racks (16 K Knoten)
        - » Mit Stand-by Menge von 4 Racks für Fail-over
    - 2 Host-Rechner
      - Verwaltung des Rechners
      - Vorbereiten der Jobs
    - I/O Racks mit RAIDs
    - Switch Racks
      - Mit Gigabit Ethernet für die Verbindung der Compute Nodes, I/O Nodes, und Host-Rechner

- IBM Blue Gene/L Überblick
  - Systempartitionierung
    - Partitionierung für 8 Benutzer



Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

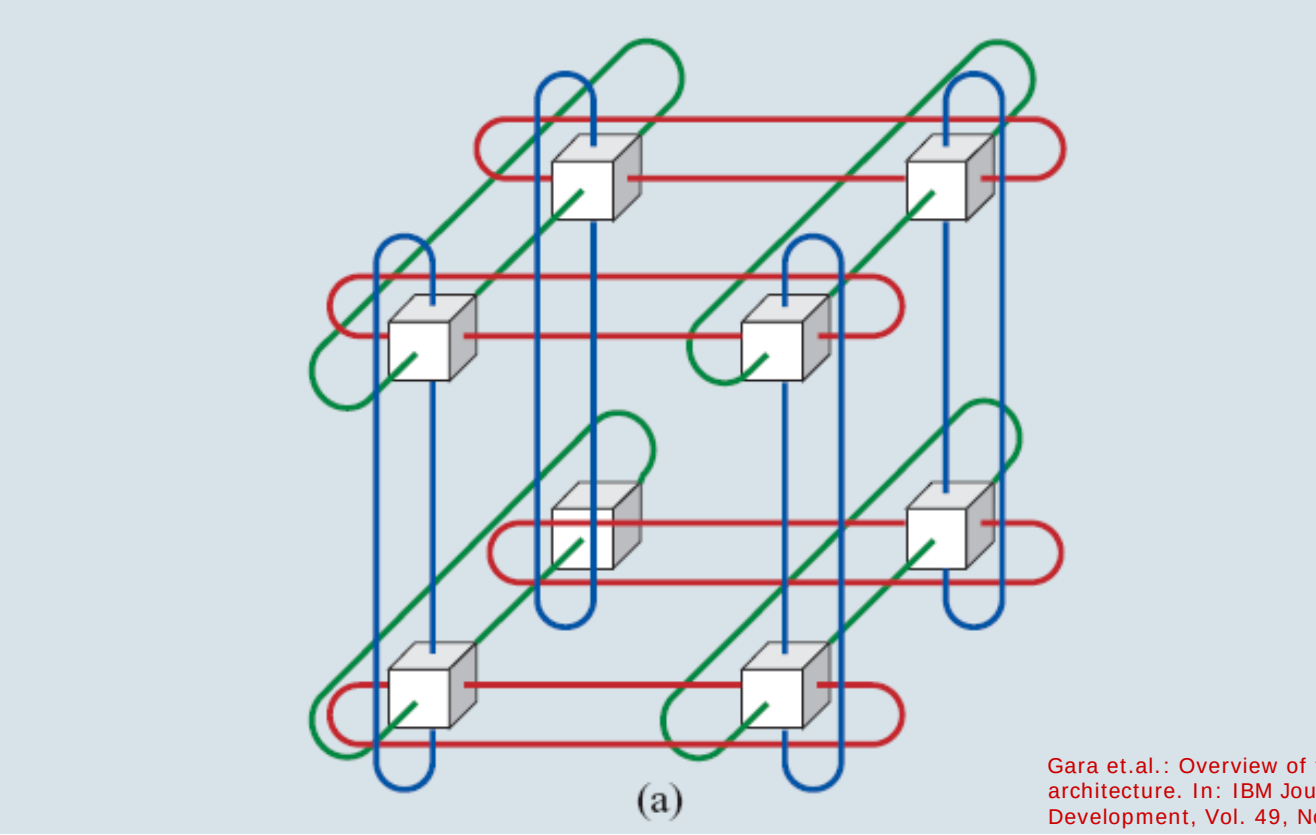
- IBM Blue Gene/L Überblick
  - BG/L Link chip



Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - BG/L Link chip
    - Ports A und B direkt mit „midplane“ verbunden
    - Ports C,D,E und F sind mit Kabeln verbunden
    - Statisches Routing, das vom Host bei der Partitionierung festgelegt wird
      - Bleibt bis zu einer Neukonfigurierung bei einer neuen Partitionierung fest
    - Jeder Link chip Port bedient 16 unidirektionale Torus Links
    - Weitere Signale für Collective und Barrier Network
    - Jede Midplane enthält 24 Link Chips
    - Jede Midplane bildet ein 8x8x8 Gitter

- IBM Blue Gene/L Überblick
  - Verbindungsnetzwerke
    - 3-D Torus (Beispiel 2 x 2 x 2 Torus)

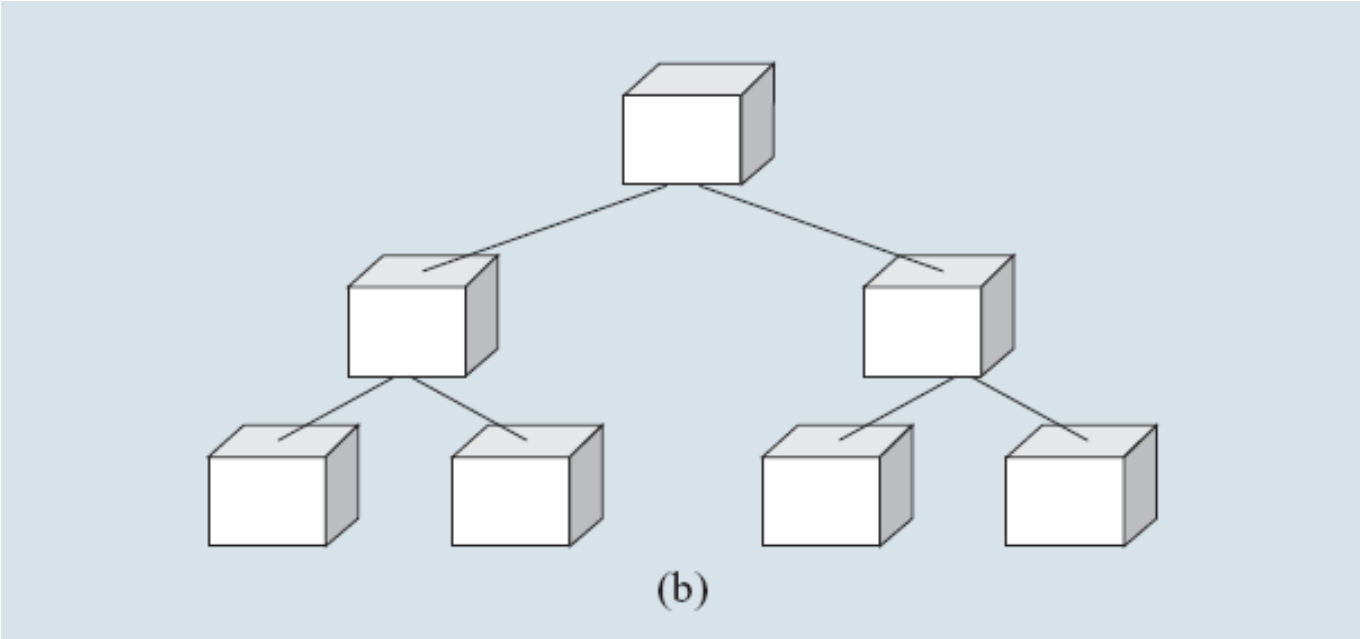


Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - Verbindungsnetzwerke
    - 3-D Torus
      - Jeder Knoten kann mit jedem Knoten kommunizieren
      - Jeder Knoten teilt seine Kommunikationsbandbreite mit Cut-through-Verkehr von anderen Knoten
      - Kommunikationsabhängige effektive Bandbreite
      - Algorithmenentwurf
        - » Möglichst lokale Kommunikation
      - Cut-Through Routing
      - Adaptive Routing
        - » Erlaubt jeden minimalen Pfad zu wählen
        - » Möglichst blockierungsfrei
        - » Dynamische Wahl der Route für die Pakete
    - Multicast-Unterstützung in jede Richtung
    - Kommunikationslatenz für für die am weitesten entfernten Knoten: 6,4µs (64Hops)

- IBM Blue Gene/L Überblick
  - Verbindungsnetzwerke
    - Collective Network
      - Erstreckt sich über gesamte Maschine
      - Daten können von jedem Knoten zu allen anderen verschickt werden (broadcast)
        - » 5µs Latenz
      - Zusätzliche Arithmetik-Reduktionsoperationen
        - » Min, max, sum, OR, AND, XOR Operationen
        - » Z.B. für globale Summation
      - Statisches Routing

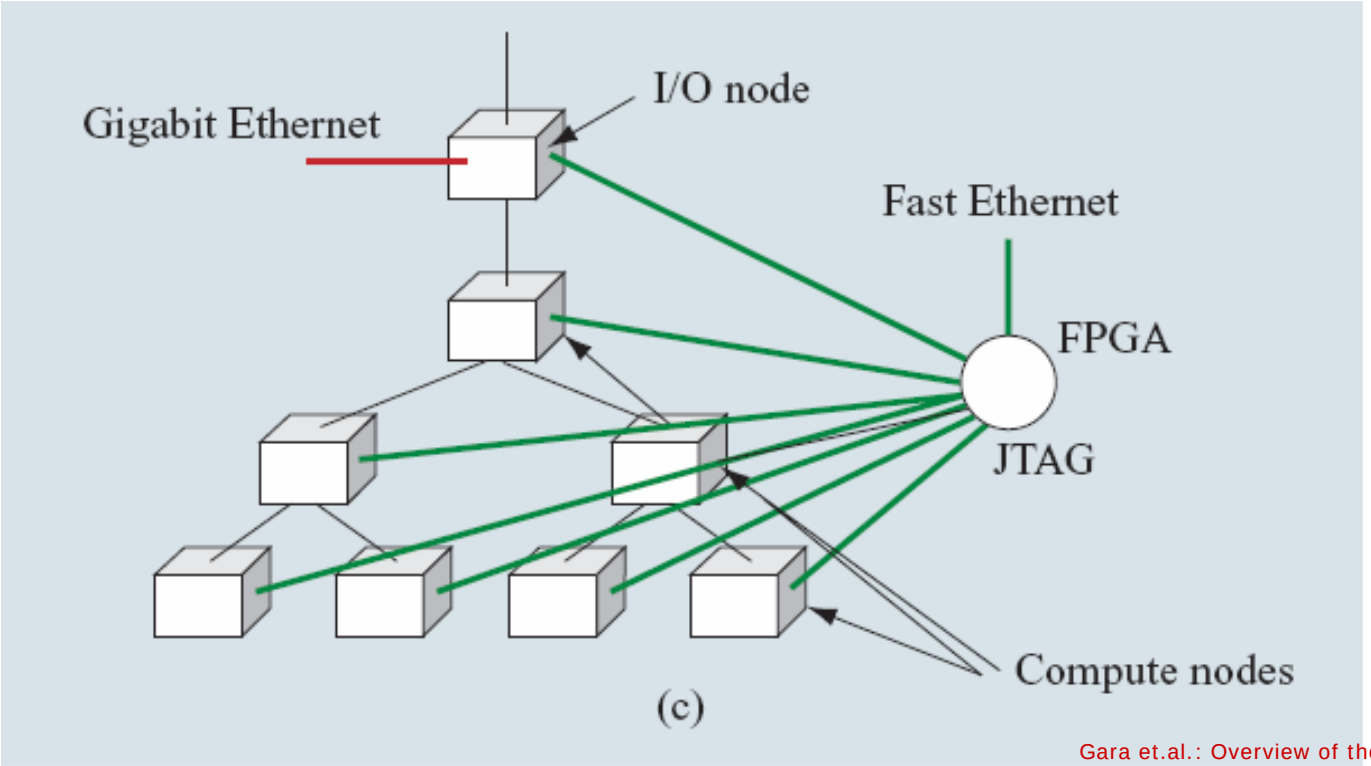
- IBM Blue Gene/L Überblick
  - Verbindungsnetzwerke
    - Collective Network



Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - Verbindungsnetzwerke
    - Barrier Network
      - Verbesserung der Latenz für globale Operationen
      - 4 unabhängige Kanäle
        - » Globales OR über alle Knoten: globaler Interrupt, wenn die Maschine oder eine Partition angehalten werden muss, z.B. für Diagnose-Zwecke
        - » Individuelle Signale werden in Hardware verknüpft und an die physikalische Wurzel eines Baums weitergeleitet
        - » Das Ergebnis-Signal wird an alle Knoten im Baum verteilt (broadcast)
        - » Globale AND-Operation mit Hilfe Inverter-Logik: globaler Barrier
        - » Round-Trip-Latenz: 1,5µs bei 64K Knoten

- IBM Blue Gene/L Überblick
  - Verbindungsnetzwerke
    - Control system network



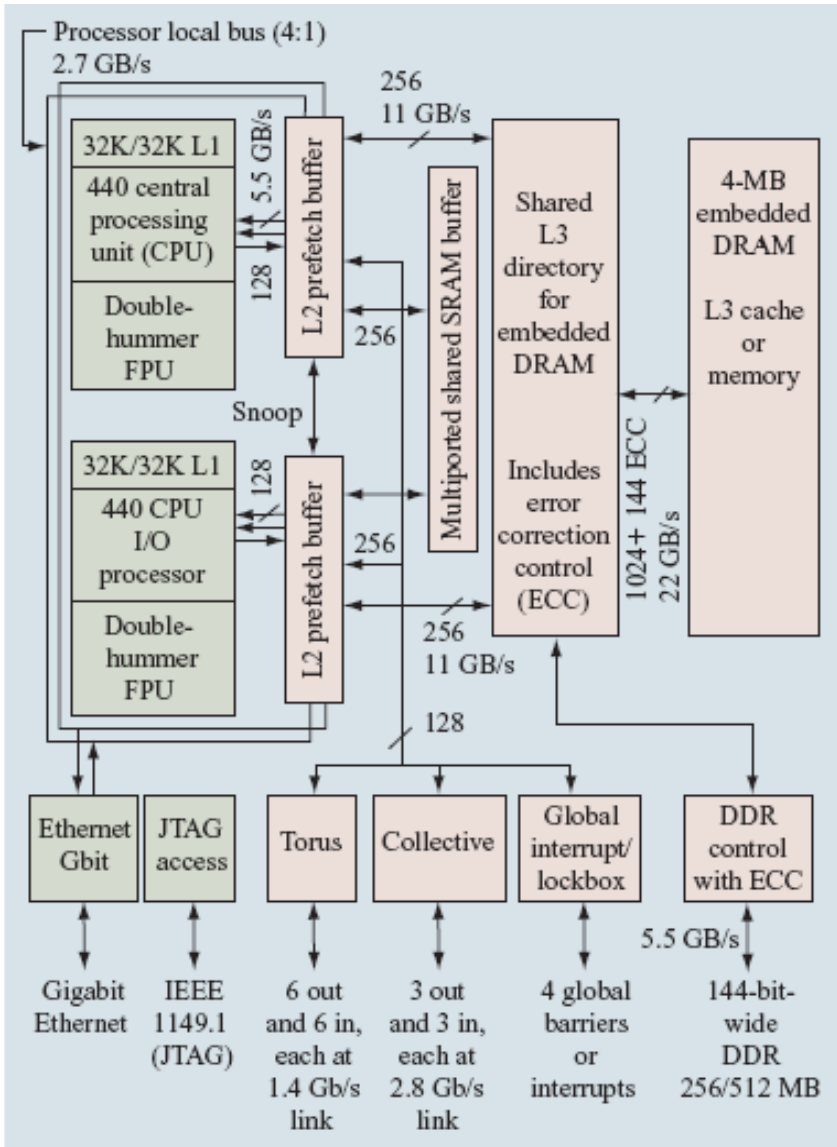
Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - Verbindungsnetzwerke
    - Control system network
      - Eine BG/L Maschine enthält eine Menge von 250000 Endpunkten in Form von ASICs, Temperatursensoren, Spannungsversorgung, Taktversorgung, Kühler, Status-Leuchtdioden, etc., die alle initialisiert, gesteuert und beobachtet werden müssen
      - Diese Aktionen werden von Service Node durchgeführt
      - Zugriff auf Endknoten über ein Intranet auf Ethernet-Basis
      - Control-FPGA übernimmt Protokoll-Umsetzung in verschiedene Netzwerkprotokolle

- IBM Blue Gene/L Überblick
  - Verbindungsnetzwerke
    - Gigabit-Ethernet
      - I/O-Knoten haben Gigabit-Ethernet-Schnittstelle für den Zugriff auf externe Ethernet-Switches
      - Verbindung zwischen I/O-Knoten und dem externen parallelen File-System sowie zum externen Host
      - Anzahl I/O-Knoten ist konfigurierbar
        - » Maximales I/O zu Compute-Node-Verhältnis ist 1:8

- IBM Blue Gene/L Überblick
  - Blue Gene/L Node
    - BLC ASIC
      - SoC, integriert die wesentlichen Funktionen eines Rechners auf einem Chip
        - » 2 PowerPC 440
        - » FP-Core für jeden Prozessor
        - » Embedded DRAM
        - » DDR Memory Controller für externen Speicheranschluss
        - » Gigabit Ethernet-Adapter
        - » Alle Puffer für die Torus-Netzwerk-Schnittstelle

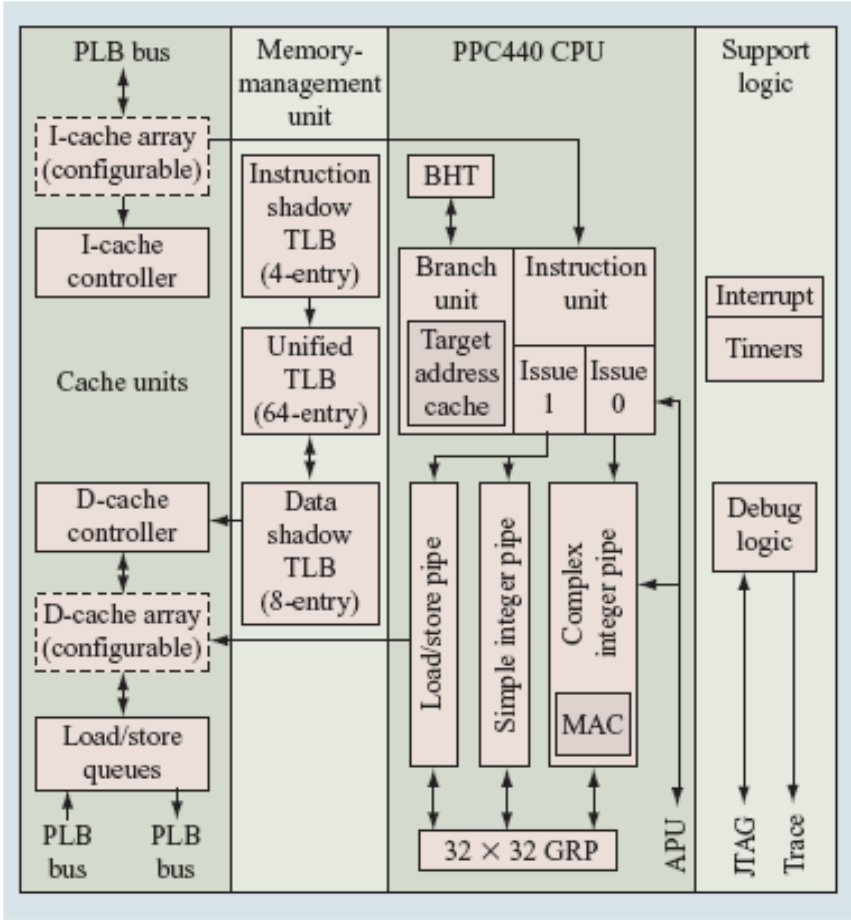
- IBM Blue Gene/L Überblick
  - Blue Gene/L Node
    - BLC ASIC



Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - Blue Gene/L Node
    - PowerPC 440
      - Taktfrequenz: 700 MHz
      - Superskalartechnik
      - 32-Bit Book-E Enhanced PowerPC Befehlssatz-Architektur
      - 7-stufige Pipeline

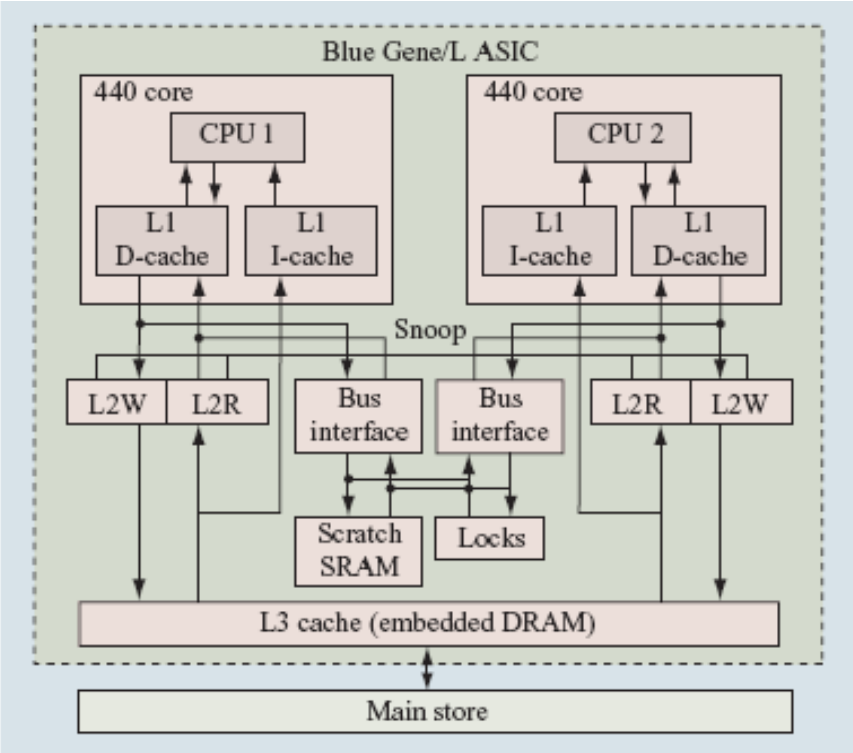
- IBM Blue Gene/L Überblick
  - Blue Gene/L Node
    - PowerPC 440



Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - Blue Gene/L Distributed Memory Architektur
    - Hierarchie:
      - On-chip Cache-Hierarchie
      - Off-Chip Hauptspeicher
      - On-Chip-Logik für Synchronisation und Kommunikation der beiden Prozessoren auf dem Chip
    - Verteilte Speiche-Architektur
      - Jeder Knoten hat 512 MB physikalischen Speicher
        - » Gemeinsamer Speicher für die beiden Prozessoren auf dem Chip
      - Insgesamt: 32 TBytes Speicher

- IBM Blue Gene/L Überblick
  - Blue Gene/L Distributed Memory Architektur



Gara et.al.: Overview of the Blue Gene/L system architecture. In: IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, pp.195-212

- IBM Blue Gene/L Überblick
  - Blue Gene/L Distributed Memory Architektur
    - Kohärenz
      - PPC440 Core keine Kohärenz-Unterstützung
        - » SW unterstützt Kohärenz auf L1-Ebene
      - L2 und L3 sind sequentiell konsistent mit Hardware-Unterstützung
      - Kein Inklusions-Eigenschaft für L1 und L2 sowie L1 und L3

- IBM Blue Gene/L Überblick
  - Blue Gene/L Distributed Memory Architektur
    - Kohärenz
      - PPC440 Core keine Kohärenz-Unterstützung
        - » SW unterstützt Kohärenz auf L1-Ebene
      - L2 und L3 sind sequentiell konsistent mit Hardware-Unterstützung
      - Kein Inklusions-Eigenschaft für L1 und L2 sowie L1 und L3
      - Communiaction coprocessor mode
        - » Ein Prozessor übernimmt Kommunikationsaufgaben
        - » Der andere Übernimmt die Berechnungen
        - » L1 Kohärenz wird auf System-Ebene mit Hilfe von Bibliotheken erreicht
      - Virtual Node Mode
        - » Knoten wird logisch in zwei Knoten mit jeweils einen Prozessor und dem halben physikalischen Speicher aufgeteilt
        - » Jeder Prozessor kann auf seinen eigenen Speicherbereich lesend und schreibend zugreifen und auf den anderen lesend
        - » Vermeidet Duplizieren von Anwendungsdaten
        - » Auf Knoten laufen zwei Anwendungsprozesse

- Literatur:
  - IBM Journal of Research and Development, Vol. 49, No. 2/3, 2005, Special Issue