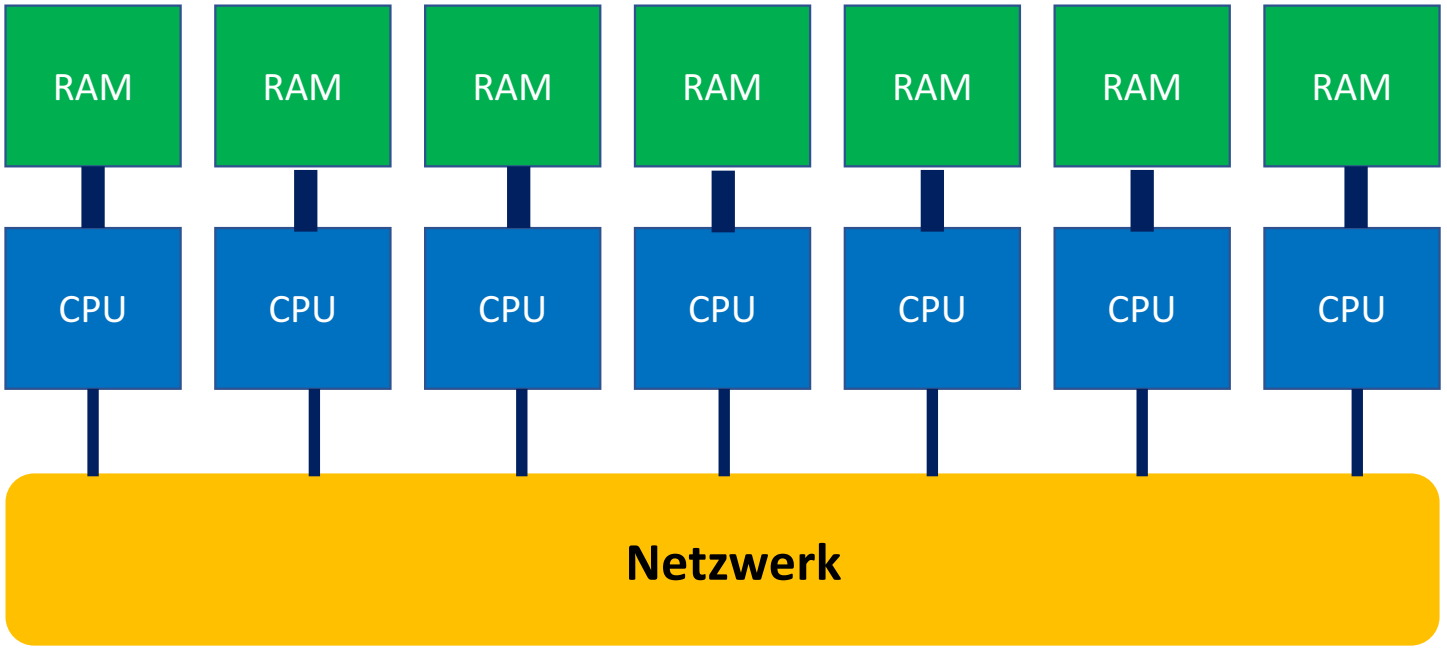
HPC-Systeme
Network

Prof. Dr. Volker Gülzow

Dr. Yves Kemp

SS 2017

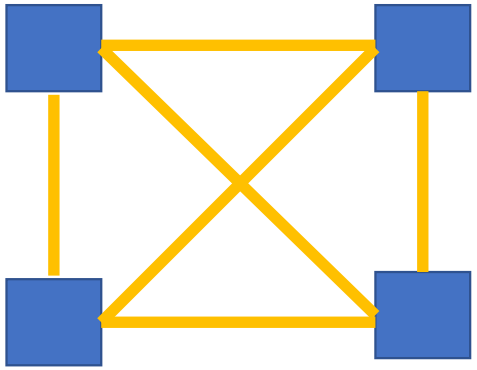
Wiederholung: Distributed Memory Systeme



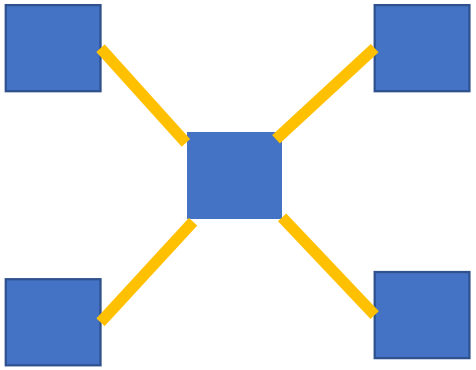
Jeder Prozessor hat seinen eigenen Speicher.
Message Passing um Daten zwischen Prozessoren auszutauschen

Interconnects: Ein bisschen Graphentheorie

- Voll vermasht: Jeder Prozessor ist mit jedem anderen Prozessor verbunden

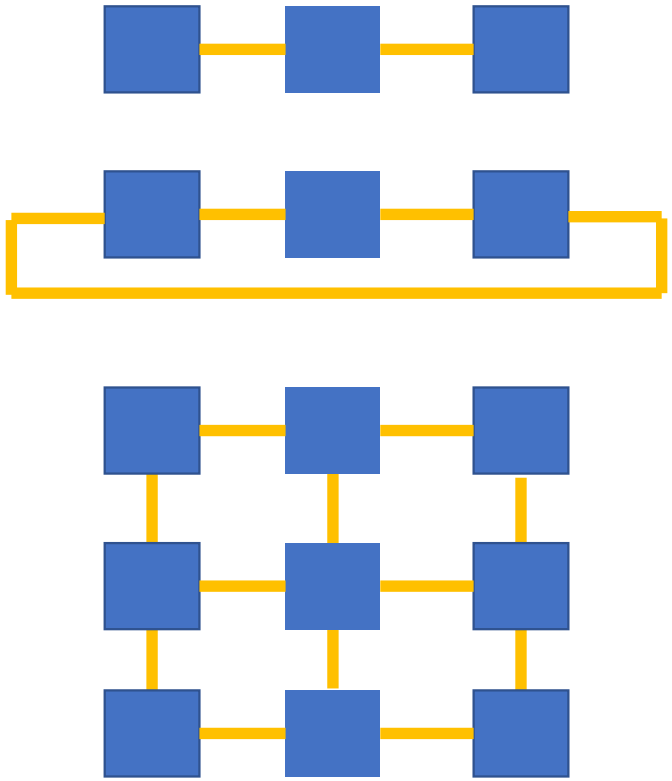


- Stern Topologie: Ein zentraler Prozessor in der Mitte, zu dem alle anderen eine Verbindung haben



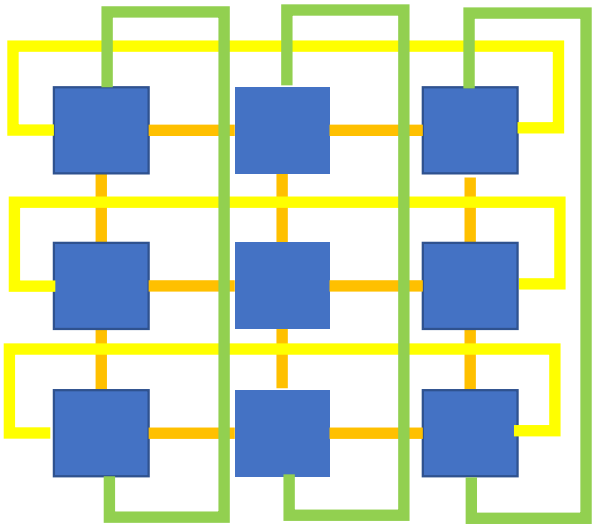
Arrays und Ringe

- Lineares Array (1-D Array)
- Ring
- Mesh Netzwerk (2-D Array)



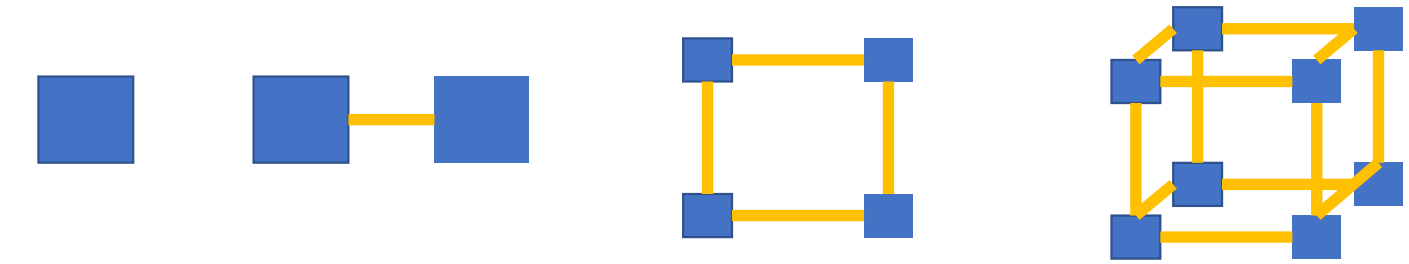
Torus .

- 2-D Torus (2-D Variante eines Ringes)
  - Die Farben dienen nur der Visualisierung: Alle Verbindungen sind gleichwertig



Hypercube

- Ein multidimensionales Mesh von Prozessoren mit genau zwei Prozessoren in jeder Dimension. Somit besteht ein d-dimensionaler Hypercube aus $p=2^d$ Prozessoren
- Hypercubes für d=0,1,2,3



Induktive Herleitung von Hypercube mit d=4

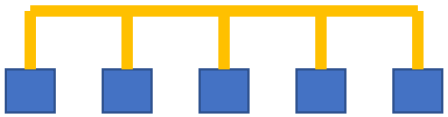
- Starte mit zwei 3-d Hypercubes
- Je zwei Prozessoren aus den beiden 3-d Hypercubes werden verbunden

Vor/Nachteile von Hypercubes

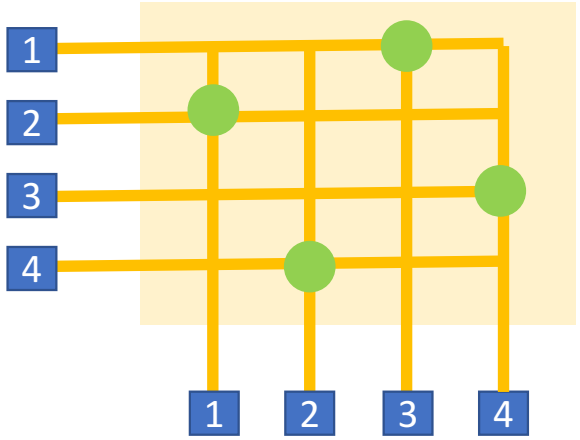
- Pro
  - Bessere Skalierbarkeit wie zB voll vermasht
  - Die Prozessoren sind gut benachbart: Nicht weiter als log(p) entfernt
  - Hohe Bandbreite
  - Wenig contention
- Nachteile
  - Benötigte Verbindungen aus dem Prozessor hängt von p ab, kompliziertes Prozessordesign
  - Mögliche Werte für $p=2^d$
  - Aufteilung der Applikation auf Hypercube schwierig (Gray-Codes, rekursives Aufteilen)

Bus / Hub und Crossbars

- Bus / Hub: Jeder Prozessor teilt den Kommunikations-Link

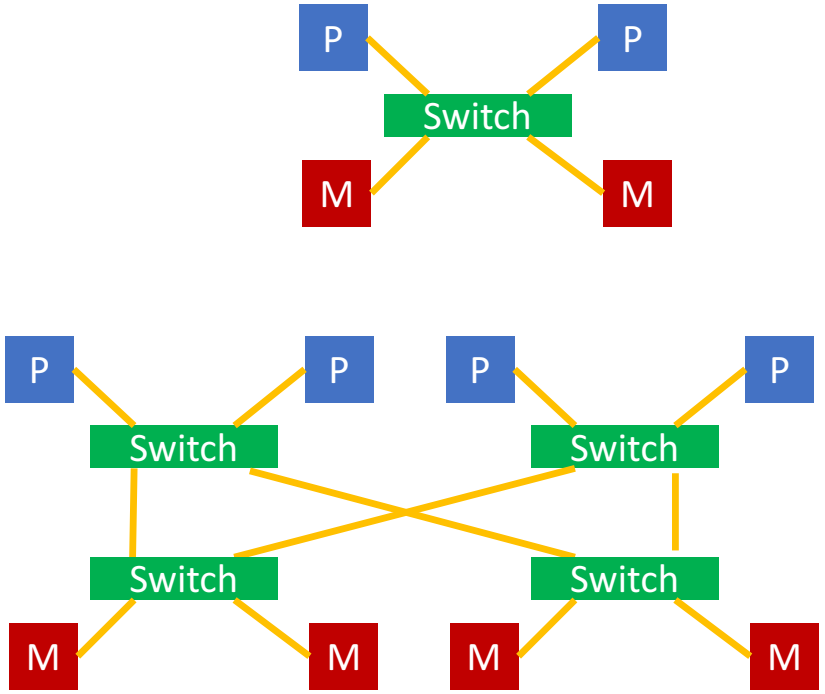


- Crossbar-Switch: Jeder Prozessor ist mit dem Crossbar Switch verbunden, der die Verbindung routet



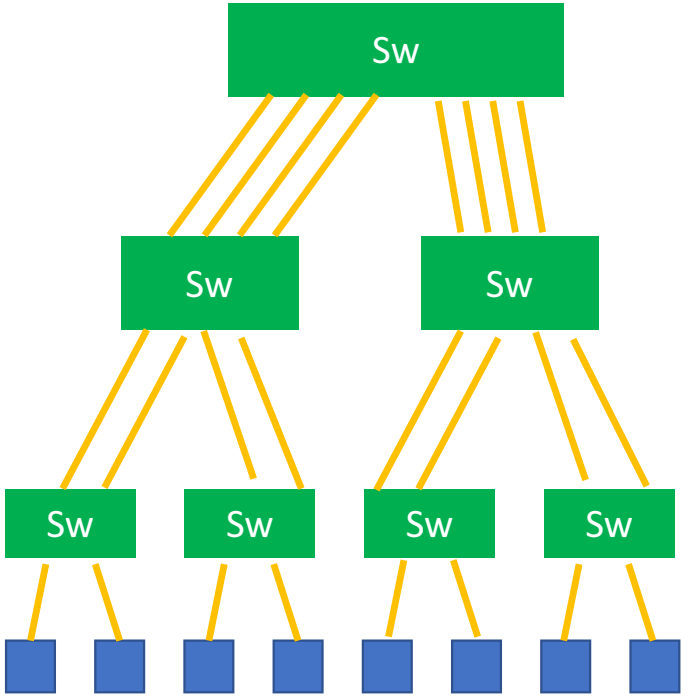
Butterfly Netzwerke

- Aus einfachen Switches aufgebaut
- Mehrere Lagen, die Anzahl der Lagen wächst mit der Anzahl der Prozessoren
- Mehrere nicht-kollidierende Wege möglich
- Uniform Memory Access



Fat Tree (Clos) Netzwerk

- Mehrere Switche, auf mehreren Ebenen
- Jeder Switch hat gleichviele Links „nach unten“ wie „nach oben“
- Die Anzahl der Links steigt mit jeder Ebene
- Volle Bandbreite zwischen den Links
- Mit jedem Switch steigt die Latenz

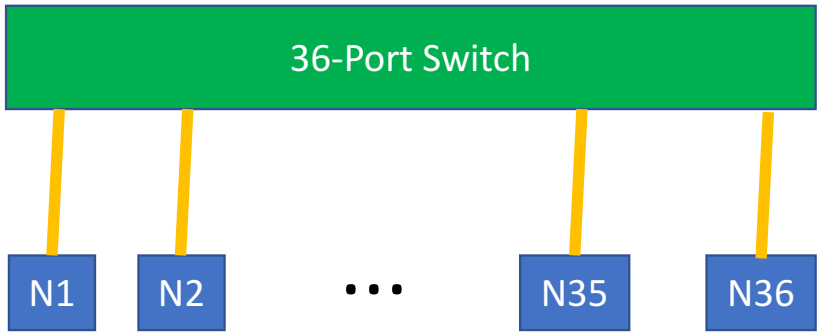


Design-Ziele

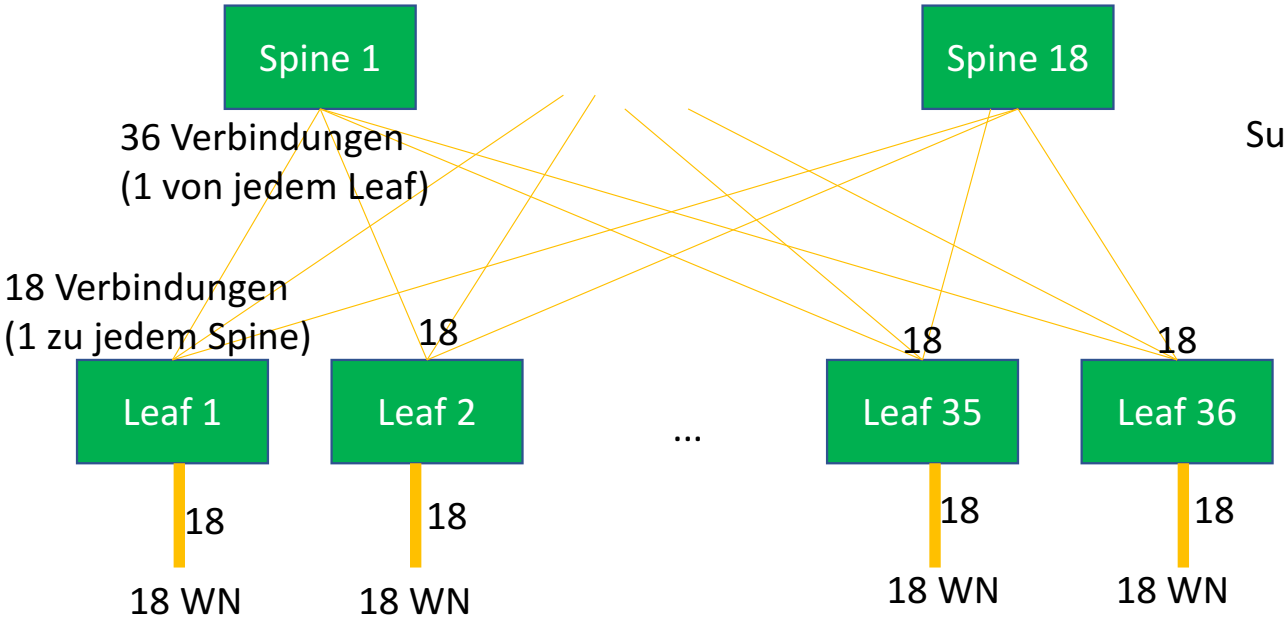
- #Links zwischen Prozessoren
  - Mehr ist besser, aber auch teurer und schwieriger umzusetzen
- Abstand zwischen Prozessoren
  - Weniger ist besser, aber skaliert typischerweise nicht gut
- Anzahl der Wege zwischen zwei Prozessoren
  - Mehr ist besser da die contention geringer wird
- Latenz
  - Weniger ist besser
- Bandbreite
  - Punkt-zu-Punkt, als auch aggregiert

Mehr zu Fat-Tree Netzwerken und InfiniBand

- InfiniBand ist vermutlich die am meisten im HPC Bereich genutzte Inter-Node Kommunikation
  - Intel OmniPath recht neu im Markt, schon einige grosse Projekte
- Basis-Bausteine der Switche sind 36-Ports ASICS (Switching Fabric)
- Einfachstes Setup:
- Skalierung?



Fat-Tree, fully-non-blocking

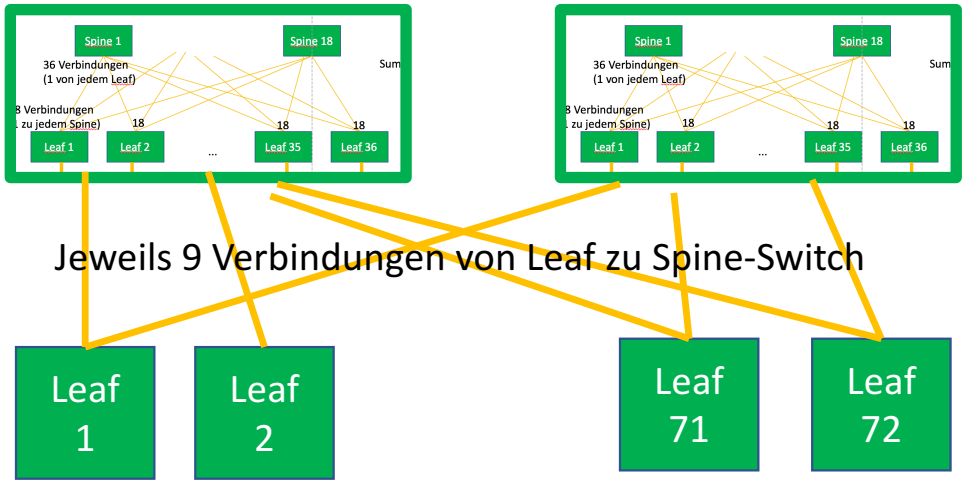


Summe: 18x36=648 Verbindungen

Summe: 36x18= 648 WN

Grössere Netze mit Fat-Tree, fully-non-blocking

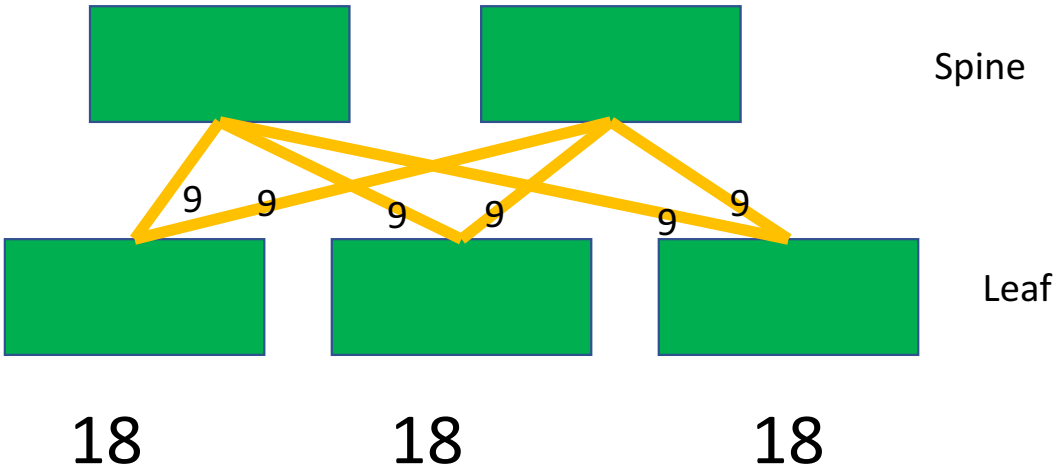
- Mindestens drei Lagen ... Aber unübersichtliche Verkablung
- Typischerweise mittels „Director Switches“ (Mellanox Namensgebung)
- Diese bauen aber auch auf den 36-port-ASICS auf
- Beispiel: 2x 648-Port Director-Switch im Spine-Level:



72x18=1296 Nodes maximal anschliessbar bei zwei 648-Port Director Switches im Spine-Level

… Und natürlich auch kleinere Setups

- 36 nodes = 1 Switche
- 37 nodes = 5 Switche
- Oder umgekehrt:
  - zwei 36-Port Switche im Spine-Level
  - =72 Nodes
  - =72/18 = 4 Leaf Switches
- Welche sinnvollen #Spine gibt es?
  - Teiler von 18: 1,2,3,6,9,18
  - Andere sind prinzipiell möglich



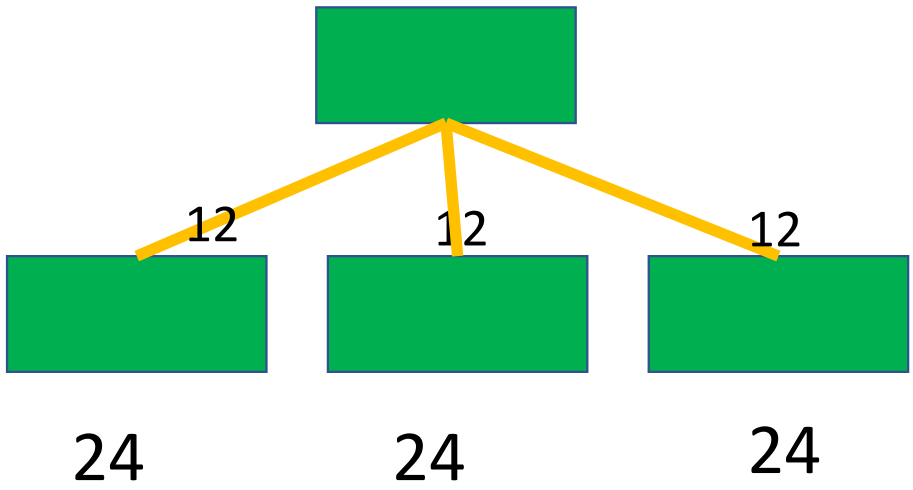
Was heisst eigentlich „fully-non-blocking“?

- Keine Contention / Kollisionen
- Ist dies notwendig?
  - Tja ... Hängt von den Applikationen ab
- Viele Cluster haben Netzwerk-Topologien mit „Blocking-Factor“
  - 2/1 ist ein beliebter Blocking Factor für Compute (Inter-Process-Communicatin)
  - Bei Storage: Man hört Aussagen wie bis „8/1 Blocking Factor kein Problem“

- Blocking Factor
  - Verhältnis Bandbreite im Leaf-Level / Bandbreite im Spine-Level
  - Kann ggf. inhomogen sein im Cluster
  - Kann auch nicht-ganzzahlige Werte annehmen

Fat-Tree, 2/1 Blocking Factor

- Alle an einem Leaf-Node angeschlossenen Nodes können „fully-non-blocking“ kommunizieren (quasi eine Insel)
- Verkehr über die Spine-Ebene unterliegt dem Blocking-Factor
- Einfachstes Beispiel
  - 1 Spine
  - 3 Leaf
  - 3 x 36 x 2 / (1 + 2) = 72 Ports



Blocking Factor, zweilagiges Netzwerk, 36-port Switches

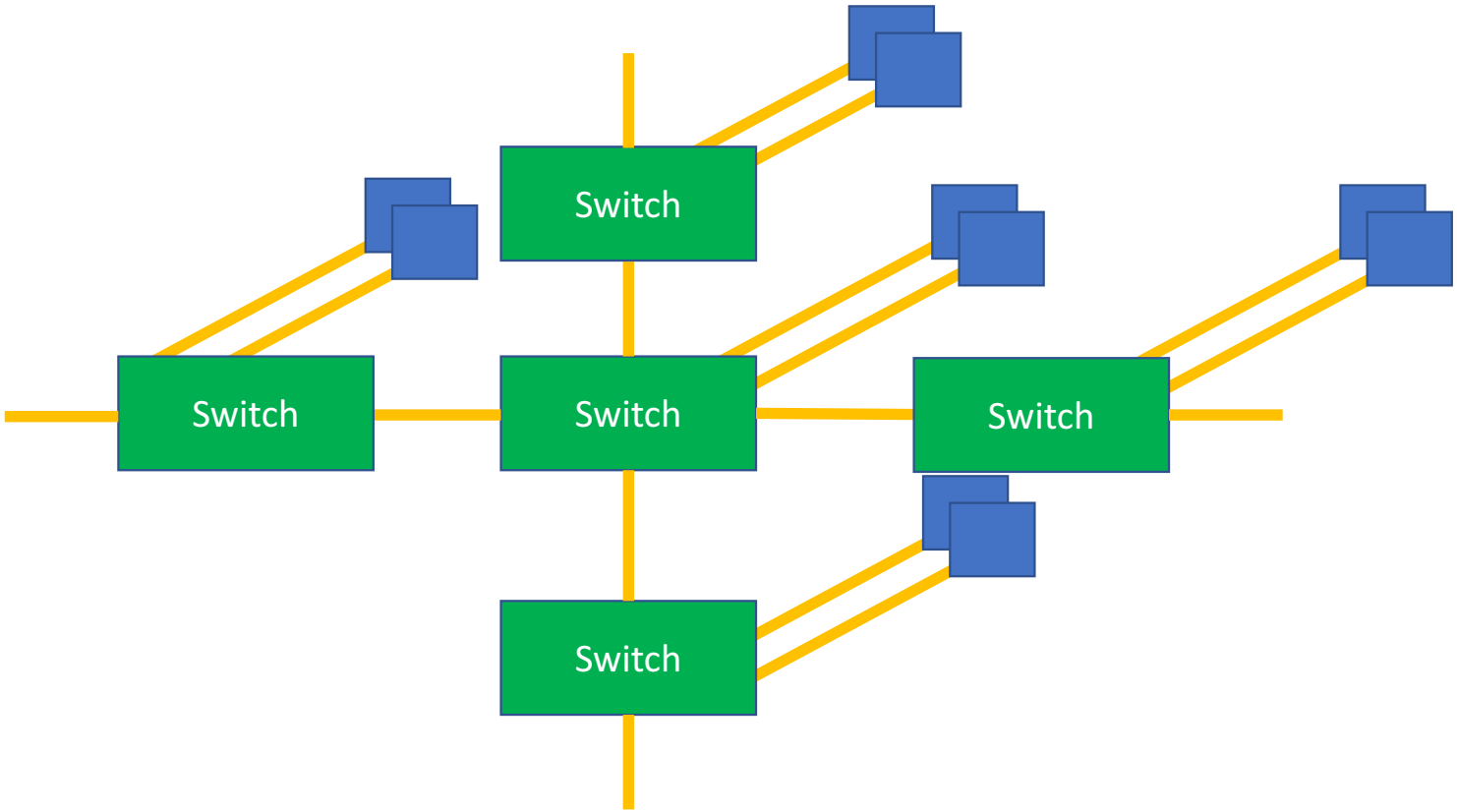
Blocking Factor	Max #Spine	Max #Port
1	18	648
2	12	864
3	9	972
5	6	1080
8	4	1152
11	3	1188
17	2	1224
35	1	1260

- Wie kommt man auf diese Werte?
- Man beachte den Sprung von 1:1 -> 2:1 Blocking Factor
  - Alle weiteren Erhöhungen des Blocking Factor bringen nur verhältnismässig wenig mehr Max#Port
- Höhere BF können interessant sein: Relativ grosse Inseln (zB 32 ports bei BF 8:1), das kann für manche Anwendungen ausreichend sein
- Scheduler muss die Topologie des Clusters kennen!

Cluster-Aufbau

- Typischerweise sind alle Nodes eines Leaf-Switches im Rack „nahe beisammen“
  - 2/1 BF: 24 Nodes -> Zb 12 Hoeheneinheiten
  - 48 Nodes = 24 HE (~24 kW) in einem Schrank, zwei Leaf-Switche
  - Kupferverkablung Node -> Leaf-Switch (0.5, 1, 2 m)
  - „Top-Of-Rack“ oder „Access Layer“ Switch
- Die Leaf-Switche sind mit den Spine-Switches verbunden
  - 2/1 BF: 6 Spine -> 18 Leaf (=9 Racks mit Nodes ... Wenn wir Storage vergessen)
  - Kupferverkablung zu umstaendlich (und zu lange Kabellaengen)
  - Glasverkablung Leaf -> Spine (Spine Switches zentral in 1-2 Racks)
  - „Core Layer“ Switches

2D/3D Torus Vernetzung: Beispiel 2D



Vorteile / Nachteile Torus

- Skalierung besser, weniger Switche notwendig
- Bessere Erweiterbarkeit
- Einfachere Verkablung, kürzere Kabel
- Fehler-toleranter

- Kommunikationswege schwieriger, ggbfs. mehr Hops

- http://clusterdesign.org/

InfiniBand: Verschiedene Generationen

	SDR	DDR	QDR	FDR10	FDR	EDR
Signaling rate (Gbit/s)	2.5	5	10	10.3125	14.0625[6]	25
Theoretical effective throughput, Gbs, per 1x[7]	2	4	8	10	13.64	24.24
Speeds for 4x links (Gbit/s)	8	16	32	40	54.54	96.97
Speeds for 8x links (Gbit/s)	16	32	64	80	109.08	193.94
Speeds for 12x links (Gbit/s)	24	48	96	120	163.64	290.91
Encoding (bits)	8/10	8/10	8/10	64/66	64/66	64/66
Adapter latency (microseconds)[8]	5	2.5	1.3	0.7	0.7	0.5
Year[9]	2001, 2003	2005	2007	2011	2011	2014[7]

https://en.wikipedia.org/wiki/InfiniBand

Mischen erlaubt

- ZB EDR abwaertskompatibel zu FDR abwaertskompatibel zu QDR

- ZB: EDR Switch Infrastruktur, aber Nodes mit FDR angeschlossen
  - Spart Kabel und Switches
  - „Nearly-non-blocking“ ... Interessante Blocking-Faktoren

Vergleich mit Ethernet

- Ethernet 10 GE ist Standard ... Vergleich zu FDR IB mit 54 Gbit/s
- 40 GE gibt es schon im Rechenzentrum, erste Sichtungen von 100 GE
  - Vergleich zu EDR IB mit 100 Gbit/s
- Welches Ethernet?
  - Normales „Natives Ethernet“
  - RDMA-over-Converged-Ethernet

http://sc16.supercomputing.org/sc-archive/tech_poster/poster_files/post149s2-file3.pdf

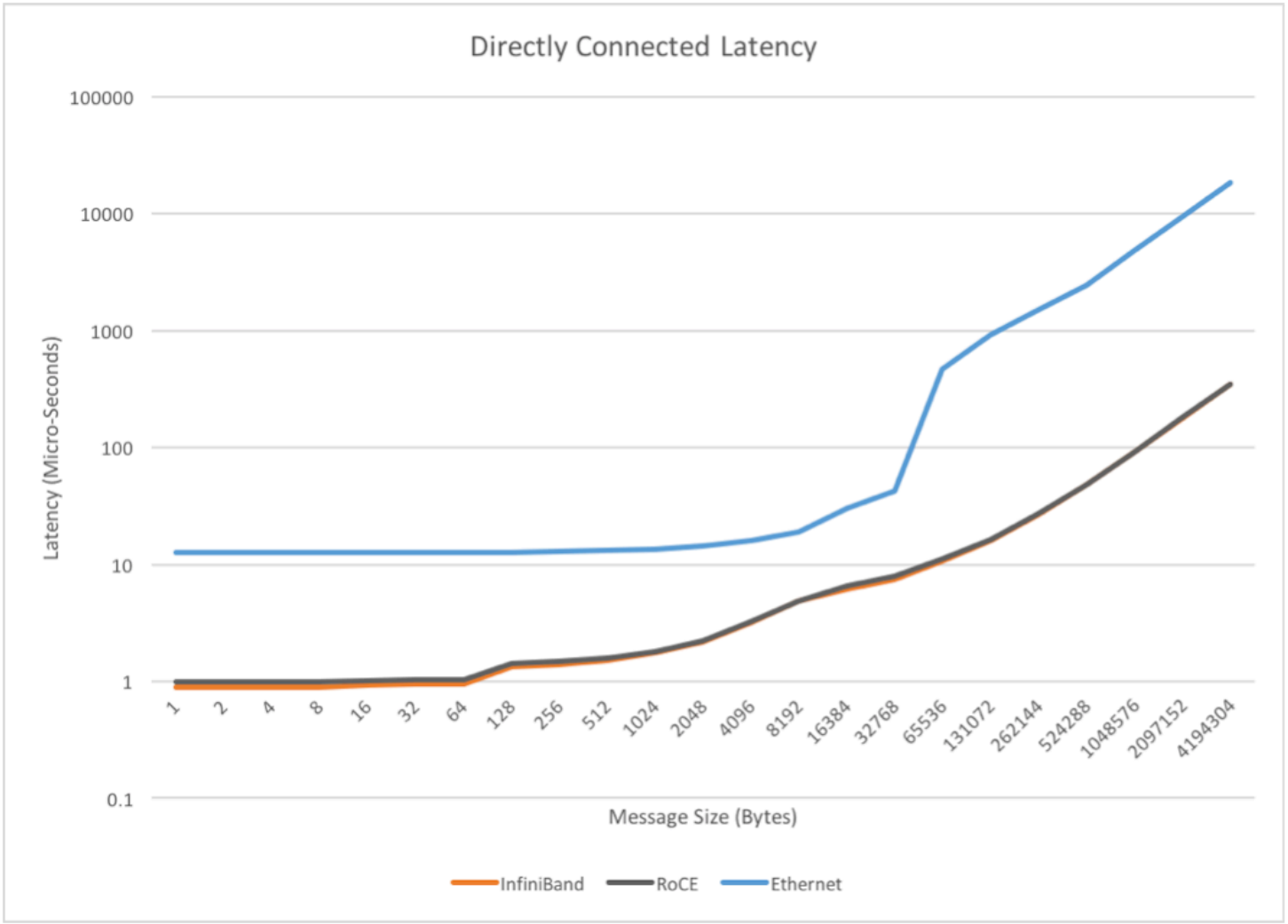
Welches Ethernet?

TABLE I. DIRECTLY CONNECTED LATENCY

Message Size	Average Latency		
	Native IB (us)	*RoCE (us)*	*Native Ethernet (us)*
8	0.892	0.988	12.662
16	0.938	1.014	12.692
32	0.95	1.026	12.718
64	0.958	1.024	12.712
128	1.344	1.418	12.78
256	1.402	1.474	12.884
512	1.518	1.584	13.104
1024	1.752	1.814	13.55

Welches Ethernet?

FIGURE I. DIRECTLY CONNECTED LATENCY



http://sc16.supercomputing.org/sc-archive/tech_poster/poster_files/post149s2-file3.pdf

Welches Ethernet?

- ... Allerdings sind nicht alle Ethernet Adapter RoCE fähig
  - Und die, die es sind, sind nicht günstiger als InfiniBand
- Wo RoCE
  - „Big-Data“ HPC-Cluster
  - Gutmütige Applikationen mit wenig inter-process-communication
  - Strategische Entscheidung für einheitliches Netzwerk
- Oder ganz Verzicht auf RDMA, dafür schnelle Verbindung: Solarflare etc.
- Dedizierte Interconnects werden definitiv weiterhin eine Rolle spielen!

InfiniBand und Alternativen

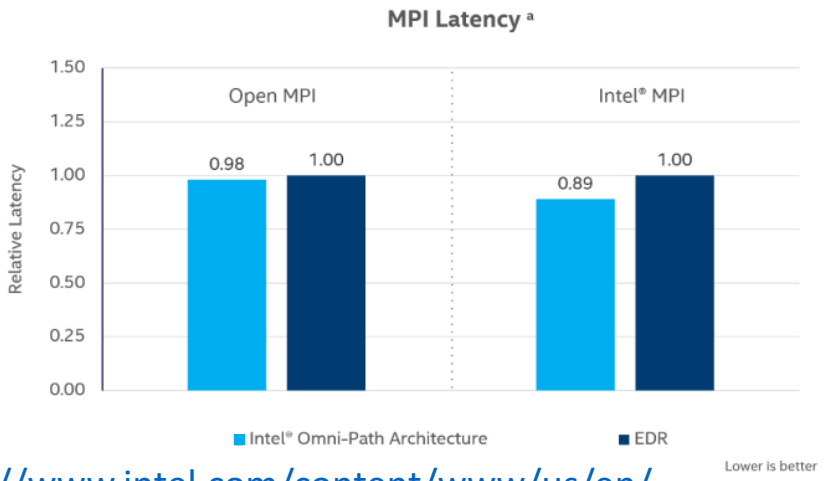
- Eigentlich nur Omni-Path als allgemein verfügbare Alternative
- InfiniBand: Aktuell (quasi) nur Firma Mellanox
- Omni-Path: Entwickelt von der Firma Intel

- Quizfrage: Ordnen Sie folgende Aussagen einem Hersteller/Technologie zu:
- A) Interconnects profitieren von einer Integration in die CPU
- B) Separate PCI-Adapter Karten ermöglichen bessere Performance zB durch Offload

Omni-Path

- Relativ neu: Verfügbarkeit erst seit Anfang 2016
- 100 Gbit/s ... Also identisch zum direkten Vergleich EDR IB
- 48-Port-Switches ... Ggbfs. Weniger Infrastruktur-Kosten als IB
- Latenz vergleichbar zu EDR IB

- „On-Load“ Strategie (auf CPU)
- Vs. „Off-Load“ bei InfiniBand

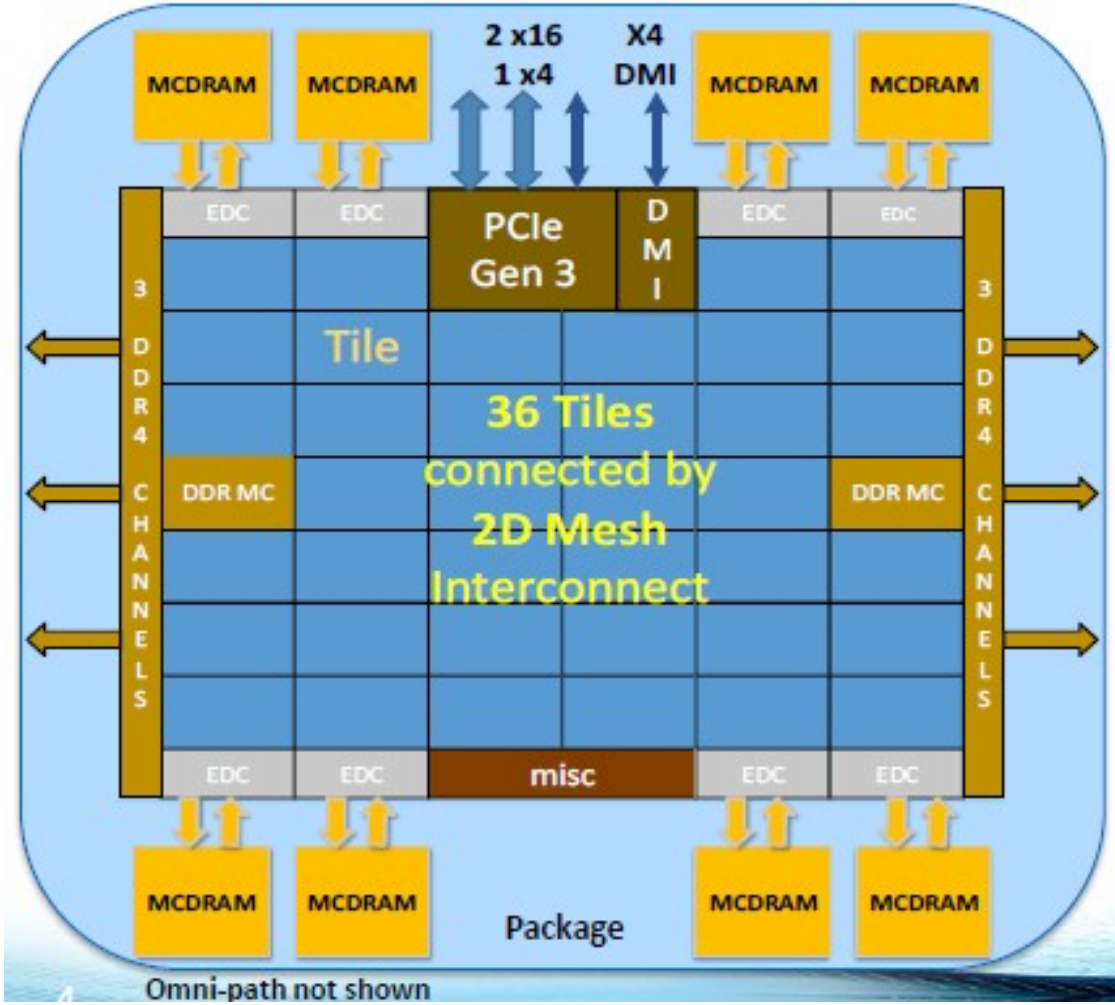


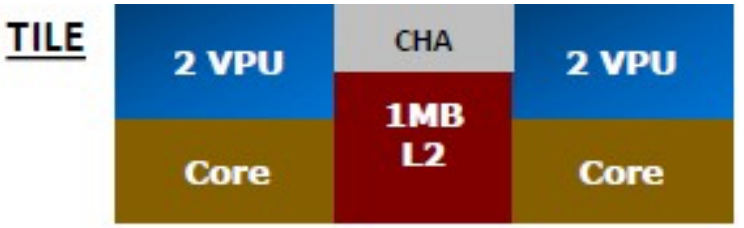
http://www.intel.com/content/www/us/en/high-performance-computing-fabrics/omni-path-architecture-performance-overview.html

Omni-Path

- Für den Hersteller interessant: Direkt auf der CPU
  - Einige Server-CPU Modelle
  - Xeon-Phi / Knights-Landing: Spezial-CPU

Knights Landing Overview





Chip: 36 Tiles interconnected by **2D Mesh**

Tile: 2 Cores + 2 VPU/core + 1 MB L2

Memory: MCDRAM: 16 GB on-package; High BW

DDR4: 6 channels @ 2400 up to 384GB

IO: 36 lanes PCIe Gen3. 4 lanes of DMI for chipset

Node: 1-Socket only

Fabric: Omni-Path on-package (not shown)

Vector Peak Perf: 3+TF DP and 6+TF SP Flops

Scalar Perf: ~3x over Knights Corner

Streams Triad (GB/s): MCDRAM : 400+; DDR: 90+

Source Intel: All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice. KNL data are preliminary based on current expectations and are subject to change without notice. 1Binary Compatible with Intel Xeon processors using Haswell Instruction Set (except TSX). 2Bandwidth numbers are based on STREAM-like memory access pattern when MCDRAM used as flat memory. Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

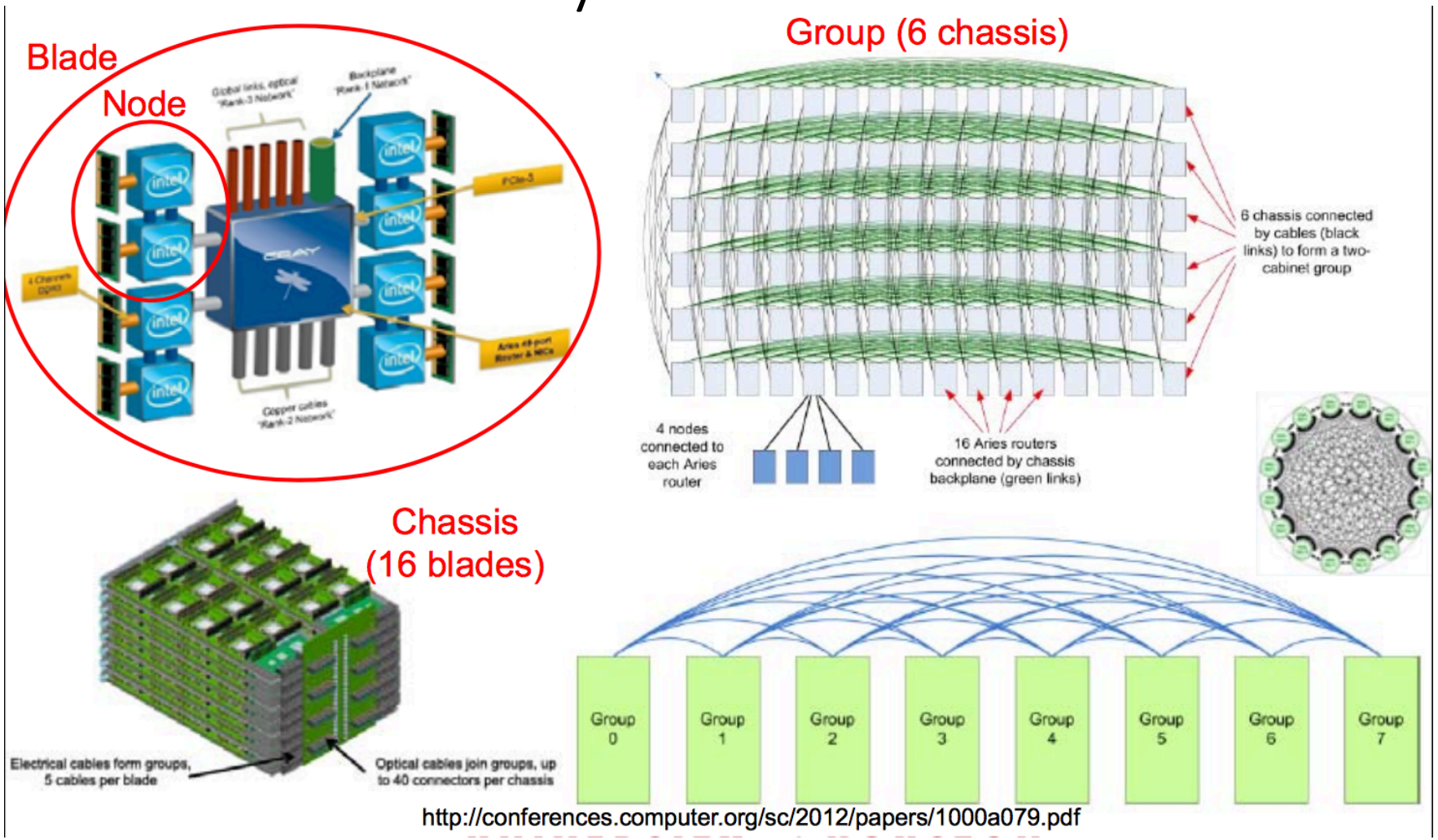
https://www.nextplatform.com/2016/06/20/intel-knights-landing-yields-big-bang-buck-jump/

Weitere Netzwerke & Topologien: Cray Dragonfly

- Exkurs: Seymour Cray *der* HPC Pionier
- Firma Cray (in mehreren Versionen) führend bei HPC Systemen und Innovation
- Aktuell mehre ganz grosse Systeme in TOP500
- Komplexer Aufbau, eigenes Netzwerk

Cray-1
Deutsches Museum
(Quelle Wikipedia)

Aufbau einer Cray XC30



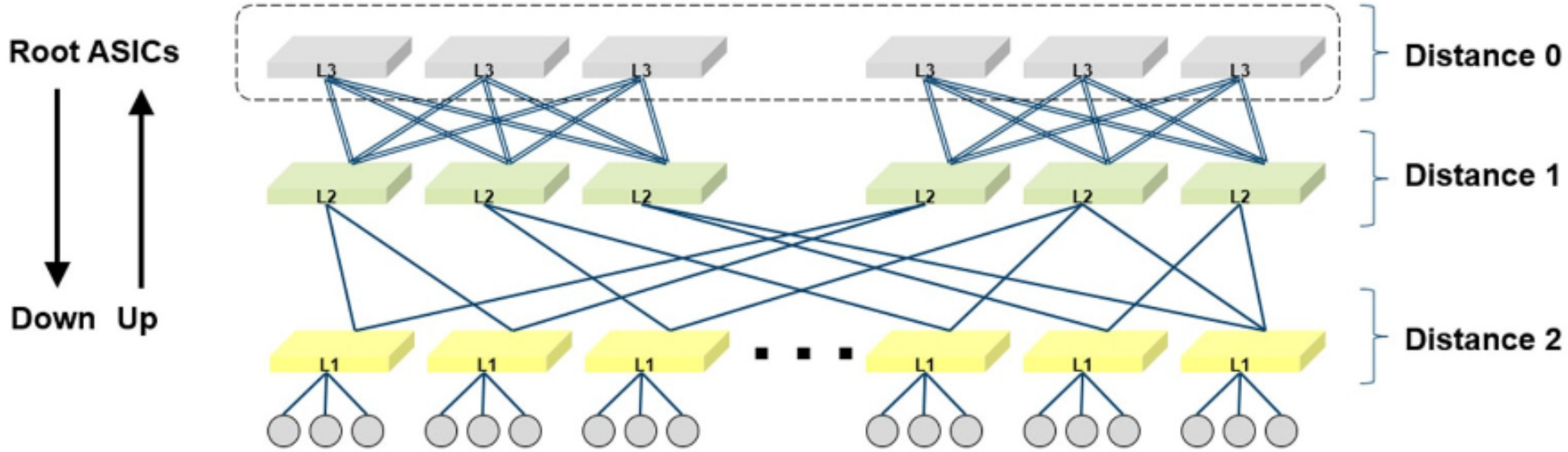
http://www2.cs.uh.edu/~johnsson/COSC6365_2014/Lecture17_Interconnect_S.pdf

Routing: Viele Algorithmen

- ... Aber erstmal Namensverwirrung:
- Routing bedeutet eigentlich: Pakete zwischen verschiedenen Subnetzen zu bewegen
- In diesem Kontext heisst Routing aber auch häufig Pakete innerhalb des gleichen Subnetzes zu bewegen ... Also eigentlich Switching

Beispiel eines Algorithmus: Up/Down

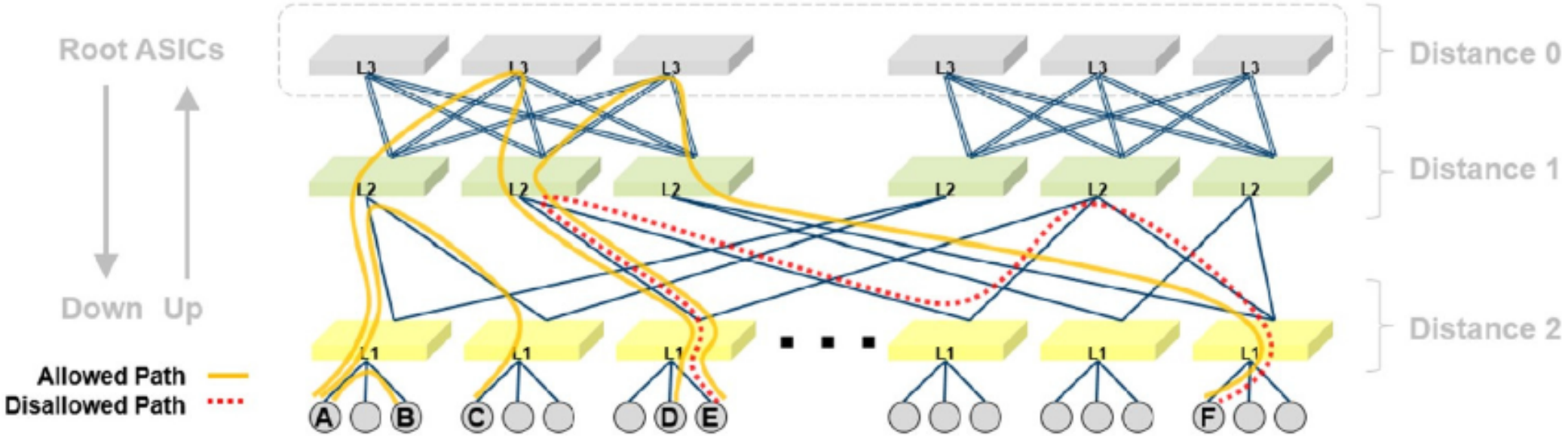
Schritt 1:
Breadth-First Spanning Tree aufbauen, und in (in diesem Beispiel) drei Ebenen kategorisieren



https://community.mellanox.com/docs/DOC-2402

Beispiel eines Algorithmus: Up/Down

Schritt 2:
Moegliche Wege suchen, wobei Wege mit Down->Up Sequenz verboten sind



Beispiel eines Algorithmus: Up/Down

Je nach Topologie können ggf nicht alle Knoten miteinander kommunizieren.
Das kann aber auch gewollt sein