

Zentralübung Rechnerstrukturen: Verbindungsstrukturen 6. Übungsblatt – Musterlösung

1 Latenzmodell

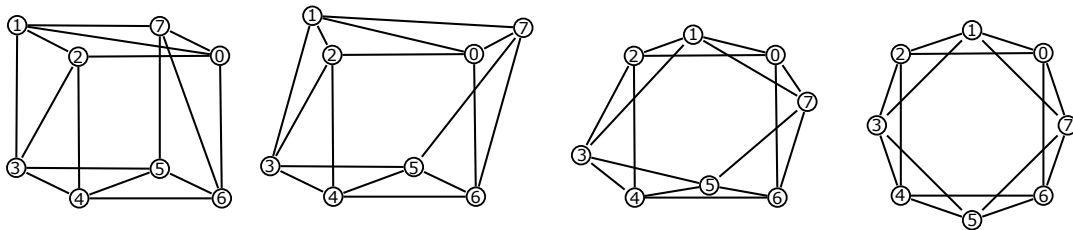
- a)
- Anzahl Schaltelemente:
3 Schaltelemente nach unten 3 Schaltelemente nach rechts, insgesamt 6
 - Zykluszeit: $\frac{1}{100 \cdot 10^6 \text{ s}^{-1}} = 10 \text{ ns}$
 - Time of flight: (kein switching OH)
Anzahl Schaltelemente $\cdot$ Zykluszeit = $6 \cdot 10 \text{ ns} = 60 \text{ ns}$
 - Transmission time:
Anzahl Phits $\cdot$ Zykluszeit = $\frac{800 \text{ bit}}{10 \text{ bit}} \cdot 10 \text{ ns} = 80 \cdot 10 \text{ ns} = 800 \text{ ns}$
 - Routing time:
Anzahl Schaltelemente $\cdot$ Routing OH + time of flight = $6 \cdot 10 \text{ ns} + 60 \text{ ns} = 120 \text{ ns}$
 - End-to-end Latenz:
 $10 \text{ ns} + 60 \text{ ns} + 800 \text{ ns} + 120 \text{ ns} + 10 \text{ ns} = 1000 \text{ ns}$
- b)
- Time of flight:
Anzahl Schaltelemente $\cdot \frac{\text{Paketgröße}}{\text{Größe eines Phits}} \cdot \text{Zykluszeit} =$
 $= 6 \cdot 80 \cdot 10 \text{ ns} = 4800 \text{ ns}$
 - Routing time:
Anzahl Schaltelemente $\cdot$ Routing OH = $6 \cdot 10 \text{ ns} = 60 \text{ ns}$
 - End-to-end Latenz:
 $10 \text{ ns} + 4800 \text{ ns} + 800 \text{ ns} + 60 \text{ ns} + 10 \text{ ns} = 5680 \text{ ns}$

2 Statische Verbindungsstrukturen

- a)
- Der Verbindungsgrad eines Knotens ist definiert als die Anzahl der Verbindungen, die von dem Knoten zu anderen Knoten bestehen. Hier also 4.

- Der Diameter oder Durchmesser ist die maximale Distanz zwischen zwei Knoten, hier in der Aufgabe 2.
- Schneidet man einen Graphen in zwei gleich große Teile und betrachtet die Menge der Kanten, die diesen Schnitt kreuzen, so bezeichnet man die Kardinalität der kleinsten dieser Kantenmengen über alle möglichen Schnitte als die minimale Bisektionsbreite. Die minimale Bisektionsbreite ist hier 6.

b) Es handelt sich hier um einen „Chordalen Ring“.



c) Es liegt Redundanz vor. Da der Verbindungsgrad jedes Knotens 4 ist und bidirektionale Leitungen verwendet werden, können bis zu drei Leitungen ausfallen und dennoch jeder Knoten von einem anderen erreicht werden. Allerdings kann beim Ausfall einer Kante der Durchmesser steigen, das heißt es könnten längere Wege notwendig sein. Anmerkung: Hier ist die minimale Anzahl von Kanten gesucht, die ausfallen dürfen, bevor ein Knoten nicht mehr erreichbar ist.

d)

	Aufgabe a)	Ring	2D-Gitter	(binärer) Baum	(n-dim) Hyperkubus
Knotenzahl	N	N	$N = n^2$	N	$N = 2^n$
Verbindungsgrad	4	2	$(2-) 4$	$(1-) 3$	$\log_2 N = n$
Durchmesser	$\lfloor \sqrt{N} \rfloor$	$N/2$	$2(n-1)$	$2(\lceil \log_2 N \rceil - 1)$	$\log_2 N = n$
min. Bisektionsbreite	6	2	n	1	$2^{n-1} = N/2$

- e)
- Jede Erweiterung benötigt die Verdopplung der Prozessorenanzahl
 - Der Verbindungsgrad der Knoten steigt bei jeder Erweiterung um 1. Rechner sind deshalb aus räumlichen Anordnungsgründen begrenzt.

3 Dynamische Verbindungsstrukturen

a) Ja.

Hier war nicht nach gleichzeitig möglichen Verbindungen gefragt (vgl. Teilaufgabe b über alle Permutationen)

b) Nein. Beweis durch Widerspruch.

Annahme: jede Permutation kann generiert werden.

Gesucht: mindestens eine Permutation, für die die Annahme nicht gilt.

- Bei einer paarweisen Mischpermutation (Kreisverschiebung), hier also Verbindung von P0 und P1 mit M6 bzw. M7 gibt es nur einen möglichen Verbindungsweg, der allerdings gleichzeitig für beide Verbindungen benutzt werden müsste.
⇒ Blockierung
- c) Schon bei zwei Verbindungen kann eine Blockierung auftreten: z.B. bei $P0 \rightarrow M6$ und gleichzeitig $P1 \rightarrow M7$
- d) Nein. Auf der einen Seite gibt es für bestimmte Paare mehrere Möglichkeiten (vgl. $P2 \rightarrow M4$), aber ebenso gibt es Paare, bei denen schon der Ausfall einer Verbindung die Weiterleitung ausschließt (z.B. $P0 \rightarrow M6$).

Beachten Sie auch die weiterführenden Erklärungen auf den Übungsfolien!

4 Leistungsmessung von Parallelrechnern

a) Nur (noch) beschränkt aussagekräftig:

- Wert ausschließlich von LINPACK-Benchmark
- Optimierung der Systeme für genau diesen Benchmark und die Top-Position der Liste
- Länder-/Kontinent-Rivalität
- Keine Aussage über reale Nutzbarkeit / Programmierbarkeit
- In Realität oft andere Anforderungen an Systeme:
 - Hoher Durchsatz an Daten, geringe Latenz
 - Spezialberechnungen mit Beschleunigern oder optimierten Prozessoren
 - Einfache und flexible Nutzbarkeit
 - Aufwand für Optimierung
 - Hoher Durchsatz an Jobs

b) Nachteile:

- LINPACK Benchmark ist > 37 Jahre alt (Top500 21,5 Jahre)
- Schwerpunkt auf Floating-Point-Operationen ($O(n^3)$), Datenbewegungen ($O(n^2)$)
- Entspricht immer weniger heutigen realen Anwendungen
- Werte sind Peak-FLOPS-Werte (normal nur 1/2 oder 2/3 von Maximum)
- Beschränkt Einsatz von neuen Architekturen
- Nutzbarkeit des Systems wird nicht gemessen
- Marketing-Tool
- Aussage über Rechner nur anhand einer Zahl

Sehr nachteiliges:

- Testet nicht die gesamte Architektur sondern nur einen Teilaspekt
 - Beschränkt die Technologie- und Architekturmöglichkeiten für HPC-Systementwickler
- ⇒ Ausrichtung der Entwicklung für diesen Benchmark
- Benchmarks über Floating-Point-Berechnungen sind immer weniger aussagekräftig
 - Datenintensive Tasks nehmen immer mehr zu

Quelle: „HPCG: One Year Later“,

c) Weitere Benchmarks:

- **Graph 500:** Big Data Computing
Cybersecurity, Medical Informatics, Data Enrichment, Social Networks, and Symbolic Networks
- **HPCG: High Performance Conjugate Gradient**
löst $Ax = b$, große lineare Gleichungssysteme
Verschiedene Kommunikationspatterns, kollektive Operationen, Speicherbandbreite, ...
- **HPC Challenge:** Verschiedene Benchmarks
- Livermore Loops
- NAS Parallel Benchmarks
- Dhrystone, Whetstone
- SPEC-hpc
- ...

d) Weitere Listen mit Top-Rechnern:

- Green 500: Energy-Aware HPC
<http://www.green500.org/>
- Graph 500: Big Data Computing
<http://www.graph500.org/>
- Green Graph 500: Energy-Aware Big Data Computing
<http://green.graph500.org/>
- Top500 HP-LINPACK vs. HPCG
<https://software.sandia.gov/hpcg/>
- ...

Beachten Sie, dass die Listen dieser Antworten nur Ausschnitte aus verschiedenen möglichen Antworten sind!

5 Vergleich von Parallelrechnern

- a)
- BlueGene/P:
 $825.500 \text{ GFLOPS} / 294.912 \text{ Kerne} = 2,80 \text{ GFLOPS/Kern}$
 - HP XC6000:
 $1.900 \text{ GFLOPS} / (101 * 2 + 10 * 8) \text{ Proz} = 6,74 \text{ GFLOPS/Proz}$
 - Achtung, dies sind sehr theoretische und vereinfachte Werte!
- b)
- BlueGene/P:
3-dimensionaler Torus, Eigenentwicklung von IBM, statisches Netz
 - HP XC6000:
Fat-Tree (Baumstruktur), Quadrics QsNet II Interconnect, dynamisches Netz, Rechnerknoten sind nicht im Netzwerk auf verschiedenen Ebenen verteilt
- c)
- BlueGene/P:
 $\sqrt[3]{294912} \approx 67$
 $\Rightarrow$ Durchmesser von 3D-Torus: $3 * 67 / 2 \approx 100$, da bei einem Torus im Vergleich zu einem Gitter der Durchmesser auf Grund der Rückwärtskanten halbiert wird. Achtung, dies ist eine theoretische Schätzung ohne Berücksichtigung des wirklichen Netzwerkaufbaus!
 - HP XC6000:
Aufsteigen im Baum bis zur Wurzel und zurück: 4
- d)
- BlueGene/P:
Netzwerk ist nicht blockierungsfrei, Bandbreitenengpässe können auftreten, die Latenz ist unterschiedlich je nach Verbindung
 - HP XC6000:
Netzwerk ist blockierungsfrei, Bandbreite von mehr als 800 MB/s, geringe Latenz
- e)
- BlueGene/P:
Prinzipiell nein.
Je nach Wegwahlverfahren können aber Probleme auftreten.
 - HP XC6000:
Nein, da ein „Dynamic Fat-Tree“ verwendet wird, bei dem jede Permutation geschaltet werden kann

- f) • BlueGene/P:
Sehr gut, das Netzwerk kann einfach um eine Ebene erweitert werden, prinzipiell unbeschränkt und ohne direkte Änderung der Netzwerkanbindung schon existierender Knoten.
- HP XC6000:
Sehr schlecht, erweiterbar um jeweils eine 2-er-Potenz, maximal 4096 angeschlossene Rechenknoten, d.h. maximal ≈ 40000 CPUs je nach Rechenknoten
- g) • Netzwerk besteht aus drei Schichten miteinander verknüpfter Switches
- Ebene 1 - Switche haben genauso viele Verbindungen zur nächsten Ebene wie angebundene Rechenknoten
- In jeder Ebene nimmt die Portzahl quadratisch zu
- ⇒ Netzwerkgröße limitiert durch die Größe (Portzahl) der Switches
- ⇒ Nach Erweiterung auf ~ 200000 Prozessoren müssten Switches mit mehr als 20000 Ports verwendet werden.
- h) • Ausdünnung der Verbindungen in Richtung Wurzel des Baumes
- Dann jedoch:
- ⇒ Keine Blockierungsfreiheit mehr
- ⇒ Durchsatz verringert sich
- ⇒ Redundanz geringer

6 Klausuraufgaben

Die Musterlösungen zu den Klausuraufgaben finden Sie auf der Homepage zur Vorlesung Rechnerstrukturen. Weitere Erklärungen finden Sie in den Folien zur Übung.