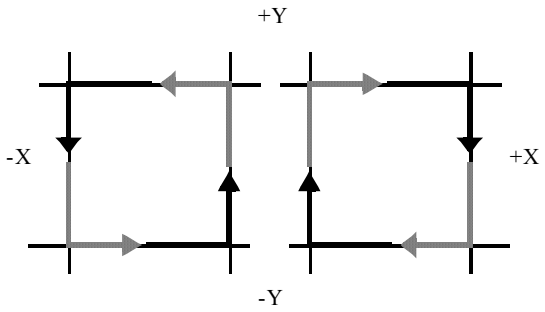
Drehrestriktionen im X,Y-Routing



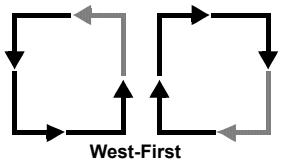
- X,Y-Routing verbietet 4 von 8 Drehungen und lässt keine Möglichkeit für adaptives Routing

Frage:

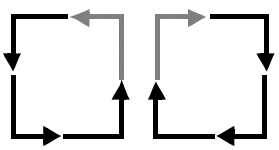
Können unter Einhaltung der Verklemmungsfreiheit mehr Drehungen erlaubt werden?

Minimale Anzahl an Drehrestriktionen

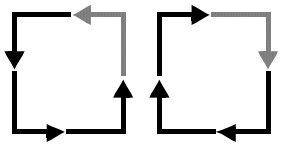
Antwort: 2 verbotene Drehrichtungen reichen aus.



West-First

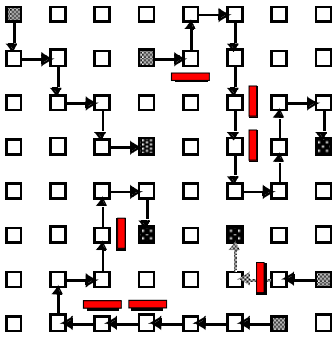


North-Last



Negativ-First

Beispiel: Legale West-First Routen



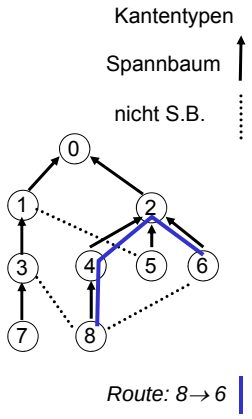
Routen um defekte Links
und Stauungen möglich

Allgemeine Lösung:

Kombination aus Drehrestriktionen und sogenannten virtuellen Kanälen

Verallgemeinerung: Up*-Down* Routing

- Gegeben **beliebiges** bidirektionales Netzwerk
- Erzeuge einen Spannbaum
- Aufsteigende Nummerierung der Knoten von Wurzel zu den Blättern
- UP - erniedrige Knotennummer; DOWN – erhöhe Knotennummer
- Weg von Quelle zur Senke nur über UP*-DOWN* Routen
  - UP Kanten, einzelne Drehung, DOWN Kanten



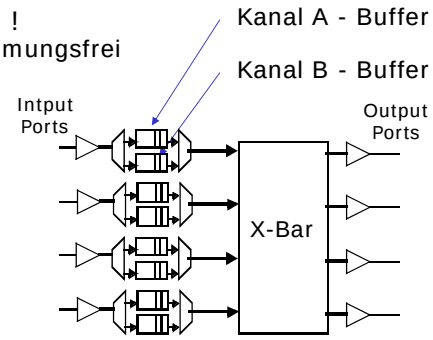
- Leistung?
  - Nicht alle Nummerierungen und Routen sind gleich gut
  - Wegverlängerung abhängig von der Topologie und Wahl des Spannbaums

Verklemmungsfreies Wormhole-Netzwerk?

Beispiel:

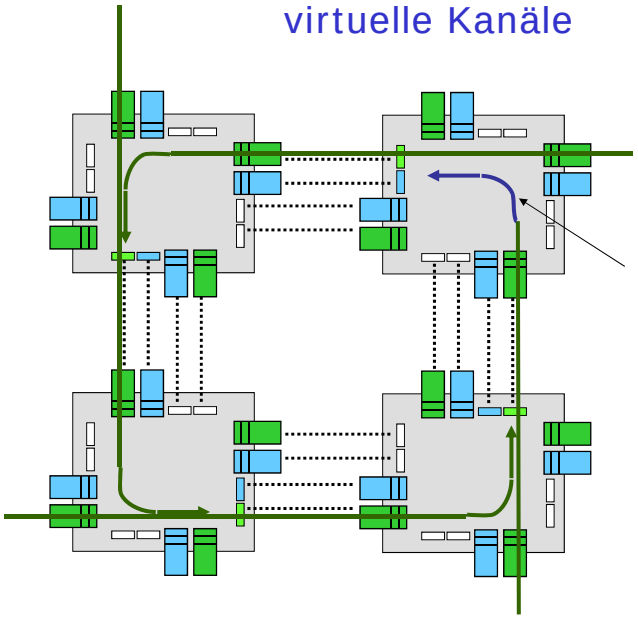
In k-ary d-cubes (Torus) funktioniert die Technik Dimension-ordered Routing nicht !
– nur k-ary d-arrays (Gitter) verklemmungsfrei

- Idee: Einführung von Kanälen!
  - Verwende mehrere “virtuelle Kanäle” zum Aufbrechen der Zyklen im Abhängigkeitsgraphen
  - Verbessert auch die Bandbreite!
  - Keine weiteren Links oder X-bars, sondern nur Bufferressourcen



- Erzeugt neue Knoten im Kanalabhängigkeitsgraphen; löscht Kanten?

Vermeidung von Verklemmungen durch virtuelle Kanäle

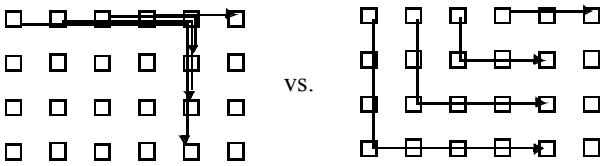


Paket wechselt vom „grünen“ zum „blauen“ Kanal

Adaptives Routing

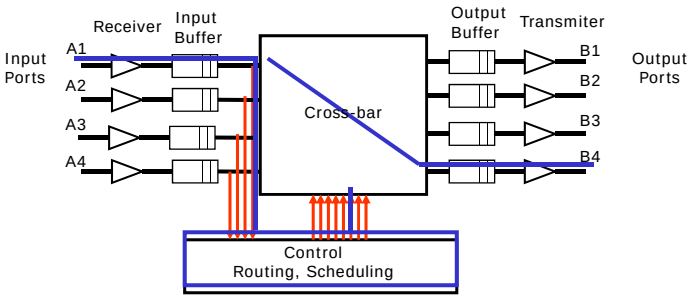
- Wichtig für Fehlertoleranz
  - Voraussetzung sind genügend alternative Wege
- Kann zur besseren Auslastung des Netzwerks führen
- Einfacher deterministischer Algorithmus führt schnell zu schlechten Permutationen

Beispiel:



- Voll/partiell adaptiv, minimal/nicht-minimal
- Kann zu weiterer Komplexität oder Anomalien führen

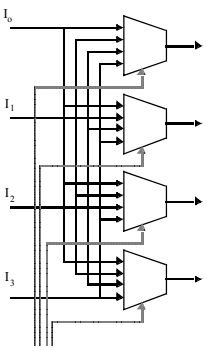
Schalterdesign

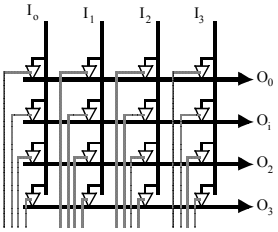


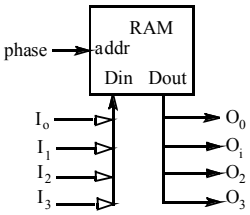
Komponenten
- Ein-, Ausgabekanäle
- Kreuzschienenverteiler
- Buffer
- Steuerlogik

Alle Permutationen sind durch einen Kreuzschienenschalter schaltbar.

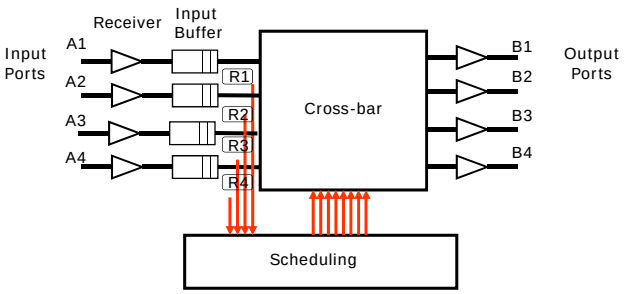
Realisierung eines Kreuzschienenschalters





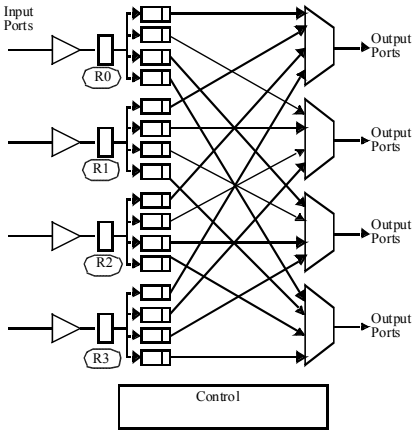


Input-Buffered Schalter



- Unabhängige Routing-Logik für jeden Eingang
- Schedulerlogik verwaltet jeden Ausgang
  - Prioritäten, FIFO, Random
- Head-of-line Blockierungsproblem

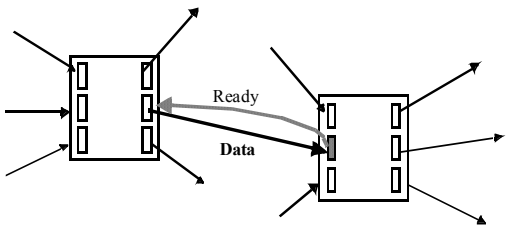
Output-Buffered Schalter



- Effiziente Nutzung der Bufferkapazitäten
  - Gemeinsam nutzbarer Bufferpool

Flusskontrolle: Link-Level

Auch auf Link-Ebene wird ein Übertragungsprotokoll benötigt.



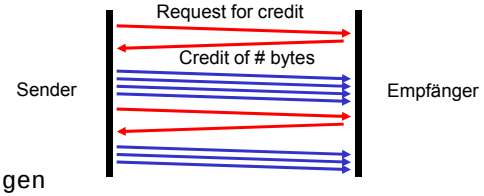
Problem:

1. Wie werden die Daten von der Quelle automatisch zum Ziel transportiert?
2. Ziel nicht verfügbar --> Quelle kann Nachrichten nicht absenden

⇒ Feed-back zwischen Empfangs- und Sendebuffer benötigt

Flusskontrolle

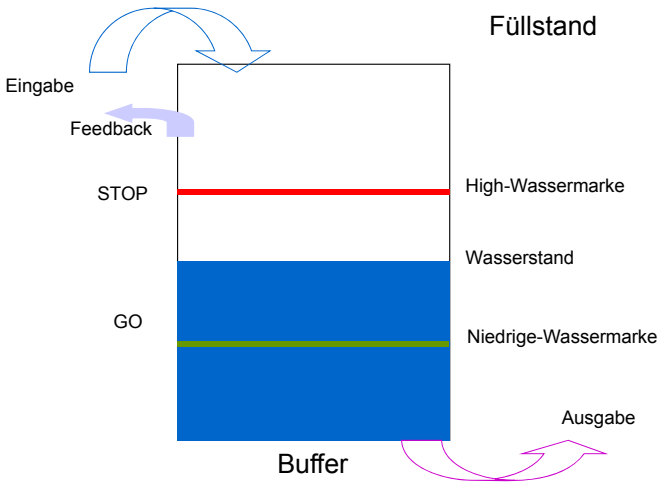
- Generelle Methoden
  - Verwerfen von Paketen
    - Erneutes versenden der Nachricht, falls keine Bestätigung eintrifft
  - Credit-based
    - Sender fragt Empfänger nach einem Credit
    - Sender darf entsprechend der erhaltenen Credit-Anzahl an Nachrichten verschicken
    - Sobald Credit aufgebraucht, neuen Credit anfordern



  - STOP/GO Benachrichtigungen
    - Buffer-Limit erreicht, STOP-Benachrichtigung versenden
    - Buffer wieder frei, GO-Benachrichtung versenden

Flusskontrolle: STOP/GO

Feed-back Methode



Zusammenfassung

- Routing-Algorithmus beschränkt die Menge der Wege in einer Topologie
  - Einfacher Mechanismus wählt Drehung bei jedem Hop
  - arithmetic, selection, lookup
- Verklemmungsfrei, falls Kanalabhängigkeitsgraph azyklisch
  - Einschränkung der Drehungen zur Eliminierung von Abhängigkeiten
  - Hinzufügen separater Kanäle zum Durchbrechen von Abhängigkeiten
  - Kombination von Topologie, Algorithmus und Schalterdesign
- Deterministisches vs. adaptives Routing
- Schalterdesign
  - input/output/pooled Buffer, Routing-Logik, Auswahllogik
- Flusskontrolle
- Reales Netzwerk besteht aus einer Vielzahl an Designentscheidungen

Zusammenfassung

- Cluster-Systeme verwenden
  - Standard Rechenknoten (SMP)
  - Spezialisierte Hardware
    - Netzwerkkommunikationsinterfaces
    - Kommunikationsnetzwerke
  - Spezialisierte Software
    - Netzwerkprotokolle
    - Message-Passing Bibliotheken

From a Single Server to a Datacenter

Requirements for Datacenters

- Servers require a defined environment
  - Save and controlled area
  - Electricity supply
  - Cooling
  - Connection to communication networks
  - Access to Storage (local and shared data)
- Quality and quantity is significant
- Challenging for hundreds of servers

- Server-side computing and cloud-computing require large farms of server systems
- Datacenters allow a high level of availability and a high degree of flexibility

Cloud Computing

- Elastic resources
  - Infrastructure on demand
  - Pay-per-use
  - Expand and contract resources
- Multitenancy
  - Multiple independent users
  - Security and resource isolation
  - Amortize the cost of the (shared) infrastructure
- Flexible service management
  - Resiliency: isolate failure of servers and storage
  - Workload: move work to other locations

Cloud Service Models

- Software as a Service (Saas)
  - Provider offers usage of centralized hosted applications to users on a subscription basis
  - E.g., email, Office 365, ..
  - Customer avoid expenses in handling own licenses, maintain software installations, and appropriate powerful hardware
- Platform as a Service (Paas)
  - Provider offers software/hardware platform for installing and running applications
  - E.g., Google's App-Engine, Microsoft Azure Paas, IBM, Fujitsu, ...
  - Customer avoid worrying about scalability of platform
- Infrastructure as a Service (Iaas)
  - Provider offers raw computing, storage, and networking
  - E.g., Amazon's Elastic Computing Cloud (EC2), Google Compute Engine, ...
  - Customer avoid buying servers and estimating resource needs

Datacenter

- Central area for housing IT equippment
  - IT systems – computer, networking, storage
  - Infrastructure
- Clear separation of IT systems and infrastructure
- Redundancy of all technical systems
  - Fail-over technics
  - Spatial separation of all infeed lines (power, water, ..)
  - Autonomous operation (e.g. emergency power supply)
- Redundancy of offered services
  - No single point of failure

- Multiple entities as a construction principal to achieve performance scaling

Datacenter - Flexibility

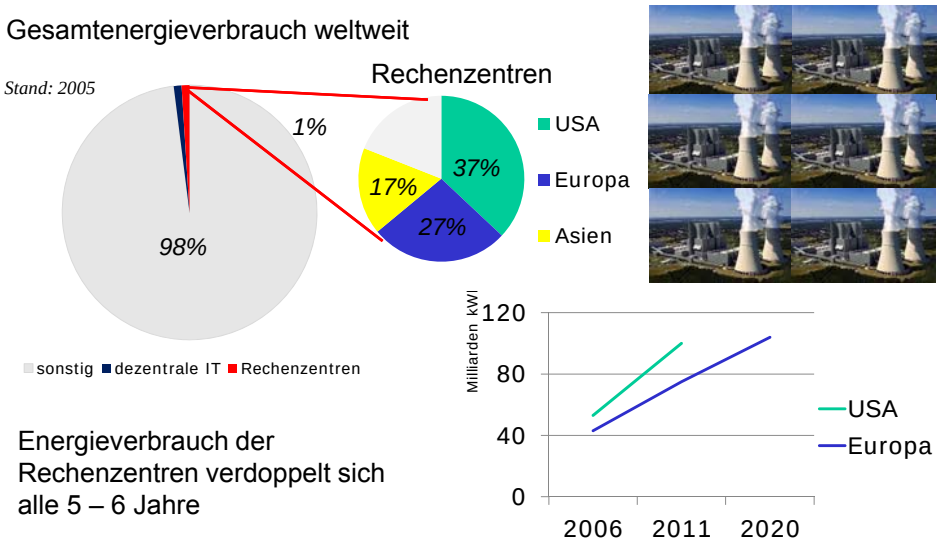
- Mix of different services possible

- Load balancing between all services
  - Sustains peak times with extreme high loads of a service
  - Boot up and shut down of services
  - High total system utilization

- Avoiding overprovisioning reduces required resources

- Shared infrastructure reduces costs

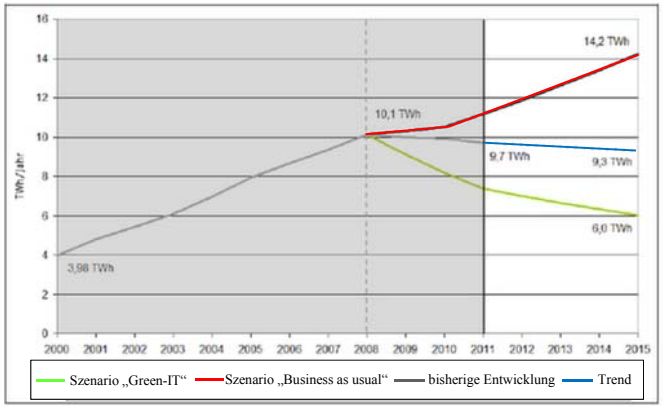
Datacenter - Energieverbrauch



Energieverbrauch der Rechenzentren verdoppelt sich alle 5 – 6 Jahre

Zeit zum Handeln

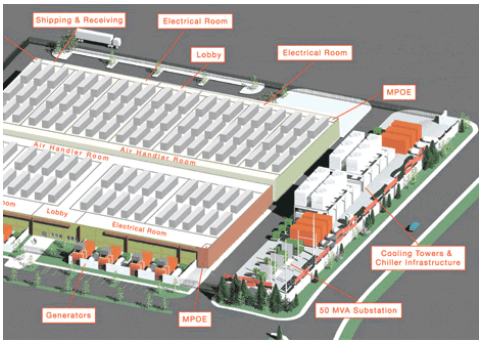
- 2,3 Mio. Server in Deutschland (2011)
- 9,7 TWh Stromverbrauch (> 1Mrd. € Stromkosten bei 0,12€/kWh)
- 1,8% des gesamten deutschen Stromverbrauchs (> 5,3 Mio.t CO_2)
- 40% des Stromverbrauchs durch Klimatisierung und USV



Quelle: Fichter/Borderstep 2010 mit Aktualisierung 2012

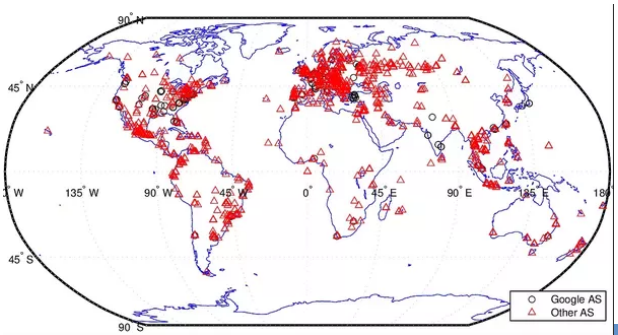
Kommerzielle Rechenzentren

- Wachstum: ca. 10.000 Server pro Monat
- Google, Microsoft und Yahoo nutzen in Zukunft Wasserkraft und freie Kühlung



Columbia River

Google Datacenter Locations



South Carolina, Goose Creek

Google

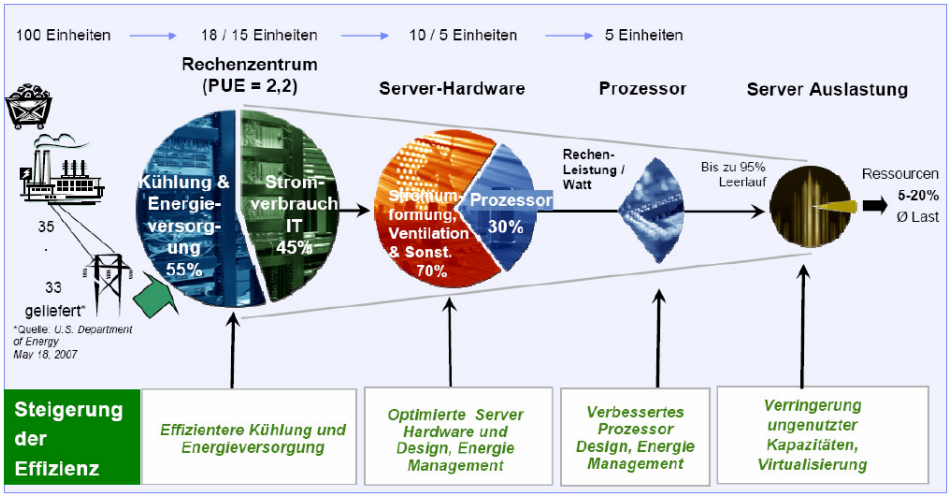
- Gartner estimates in 7/2016: 2.5 million servers
- Year 2010
  - ca. 2.26 Million MWh electrical power consumption
  - ca. 1.5 percent of world wide electrical power
- Year 2011
  - datacenter in Finland
  - 200 Million Euro investment in first phase
  - Former paper mill with a tunnel to the Gulf of Finland
  - Wind park for electrical power generation
- Year 2017
  - Datacenter in Eemshaven, Netherlands
  - 600 Million Euro investment
  - Wind park, photovolaics

Stromverbrauch

- Rechenzentrum mit 30.000 kW (260 GWh p.a.)
- entspricht Stromverbrauch einer Stadt mit 69.000 Haushalten (z.B. Stadt Paderborn)

=

Wo geht es verloren?



Green500 (6/2018)

Rank (Top500)	Site	System	Cores	RMAX [TFLOP/s]	Power [kW]	Power Efficiency [GFlops/Watt]
1 (259)	RIKEN, Japan	ZettaScaler 2.2, Xeon D 16C 1.3 GHz, IB EDR, **PEZY-SC2**	794,400	842.0	50	**17.009**
4 (149)	NVIDIA Corp., USA	DGX-1, **Xeon E5** 20C 2.2 GHz,IB EDR, **Tesla V100**	22,440	1,070.0	97	**15.113**
5 (1)	Oak Ridge Nat. Lab., USA	Summit, **POWER9 22C** 3.07GHz, **Tesla V100**	2,282,544	122,300	8,806	**13.889**
6 (19)	Tokyo Inst. Of Tech., Japan	TSUBAME, **Intel E5 14C** 2.4 GHz, OPA, **Tesla P100**	135,828	8,125	792	**13.704**
8 (5)	AIST, Japan	ABCI, FSC CX2550, **Intel Gold** 20C 2.4GHz, EDR, **Tesla V100**	391,680	19,880	1,649	**12.054**
9 (255)	Barcelona SC Center, **Spain**	MareNostrum P9, POWER9, EDR, Tesla V100	19,440	1,018	86	**11,865**
23 (2)	Nat. SC in Wuxi, China	**Sunway 260C**, 1.45GHz	10,649,600	93,014.6	15,371	**6.051**
24 (12)	Joint CTR Adanced HPC, Japan	FSC CX1640, **Intel Xeon-Phi** 68C 1.45GHz, OPA	556,104	13,554	2,719	**4.986**
26 (439)	Qingdao Nat. Lab., China	Inspur **Xeon E5** 2690v4 12C 2.6, Omni-Path	23,920	771.3	162	**4.761**

Source: top500.org/green500

Was tun?

- Effizientere Programme
  - Bessere Algorithmen, dadurch weniger Rechenoperationen
- Stromsparende Rechnersysteme
  - Geringere Frequenzen, dadurch aber mehr Parallelität in der Anwendung erforderlich
  - Netzteile, Spannungswandler
- Effizientere Peripherie
  - Kühlsysteme, Lüfter
  - Massenspeicher, Kommunikationsnetzwerke
- Höhere Auslastung der Systeme
  - Zentralisierte Dienste und intelligentes Ressourcenmanagement

Viel Erfolg bei der Prüfung!

PC² bietet

- SHK Stellen (Forschungs-, Industrieprojekte, …)
- Masterarbeiten
- …