Matching von Produktdaten in der Cloud

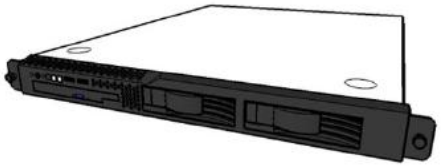
Dr. Andreas Thor
Universität Leipzig

Cloud Computing

- “Cloud computing is using the internet to access someone else's software running on someone else's hardware in someone else's data center” (Lewis Cunningham)

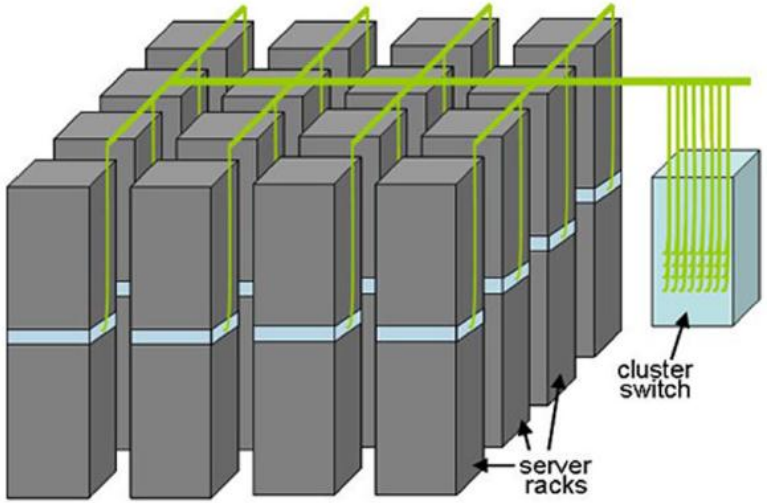
- Anbieter
  - Externe Bereitstellung von IT-Infrastrukturen sowie Applikations-Hosting über das Internet (bzw. Intranet)
  - Beispiele: Google, Amazon, …
- Nutzer
  - Verwendung der Ressourcen eines Data Centers
  - Beispiele: KMUs, Startups, …

Aufbau eines Data Centers



Server
- CPUs
- DRAM
- Disks

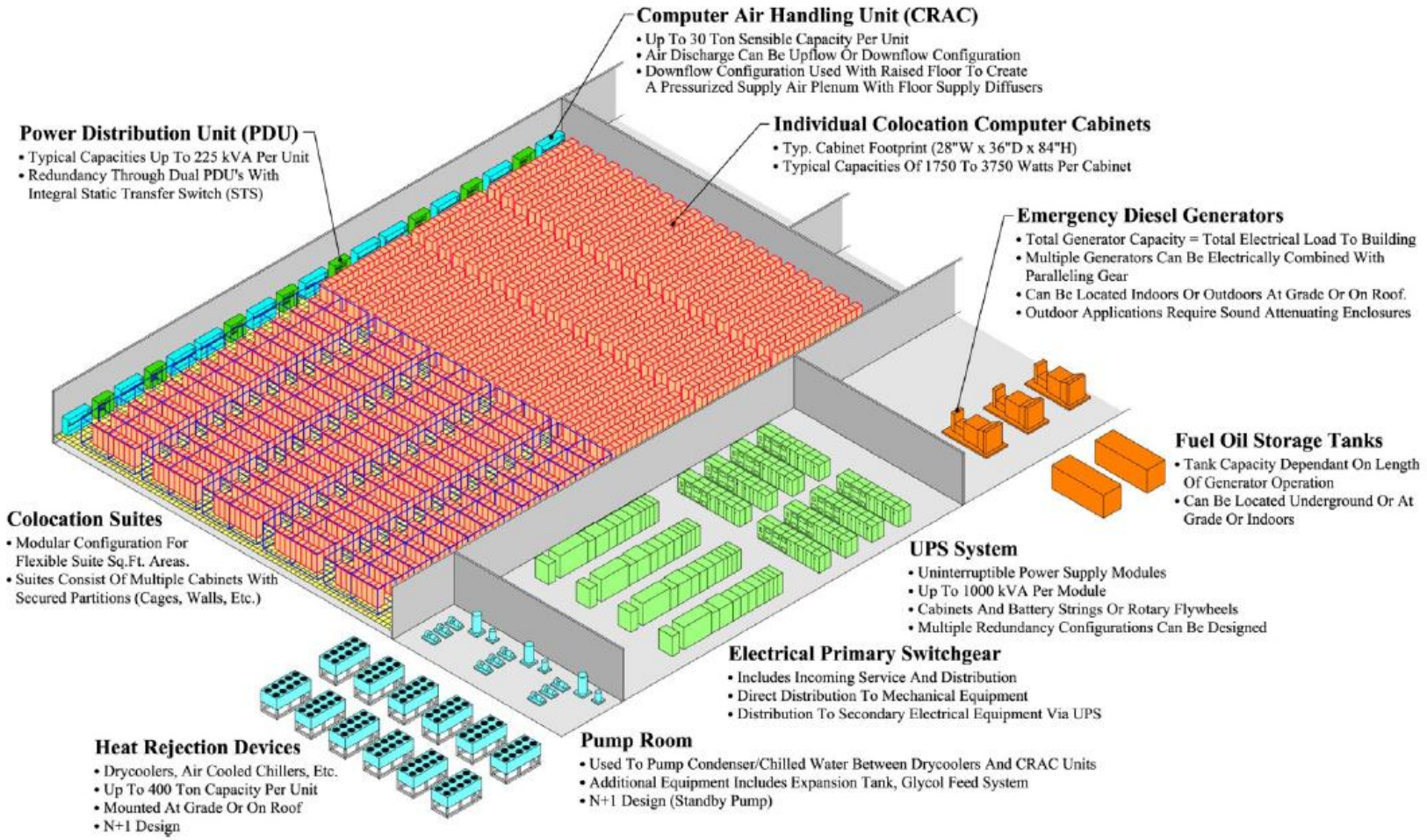
Rack
- 40-80 Server
- Ethernet Switch



Cluster

Quelle: Barroso and Hölzle: *The datacenter as a computer*: An introduction to the design of warehouse-scale machines. Morgan & Claypool, 2009

Aufbau eines Data Centers



Quelle: Barroso and Hölzle: *The datacenter as a computer*: An introduction to the design of warehouse-scale machines. Morgan & Claypool, 2009

Data Center: Ökonomische Nutzung

- Große Data Centers kostengünstig pro Server
  - ≈ Faktor 5 bei Vergleich 50.000 Server vs. 1.000 Server
- Virtualisierung ermöglicht (nahezu) freie Standortwahl nach ökonomischen Gesichtspunkten
  - Strompreis, Löhne, Steuern, ...
- On-Demand-Nutzung durch Kunden
  - Flexible Bezahlung nach Nutzungsumfang
- Nutzungsarten
  - Infrastruktur: Mieten von Ressourcen (CPU, Speicher, …)
  - Plattform: Bereitstellung/Ausführung eigener (Web-)Anwendungen
  - Service: Nutzung von Software als Web-Dienst (z.B. Mail)

Produkt-Matching Beispielszenario

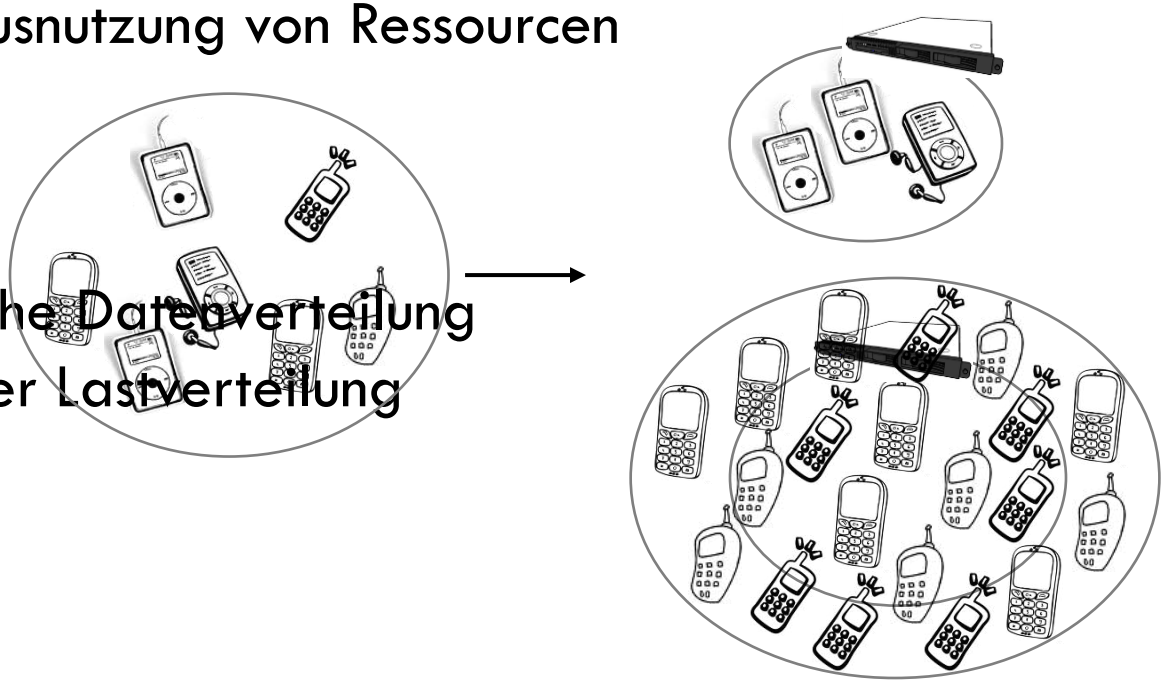
- Täglicher Import von Produktangeboten inkl. Matching
  - Dauer: 1h pro Tag
- Kostenrechnung für Amazon EC2
  - Mieten einer virtuellen Maschine
  - „High-CPU On-Demand Instance Extra Large (Europe)“ für $0,76 pro Std
  - 7 GB RAM, 8 virtuelle Kerne („2.5x1,7GHz“), 1.7 TB Festplatte
  - 1 Stunde pro Tag x 365 Tage pro Jahr x 3 Jahre ≈ 650 Euro
- Vergleich mit Kosten für eigenen Server

„Elastisches“ Produkt-Matching

- Variabler Aufwand für Produkt-Matching, da abhängig von
  - Größe des Datenvolumens
  - Anzahl der eingesetzten Match-Verfahren
  - Komplexität/Laufzeit der Match-Verfahren (Algorithmen)
- Beispiel
  - 5-fache Datenmenge führt in etwa zu 25-facher Arbeitslast (bei paarweisen Produkt-Vergleichen)
  - Aus 1h werden 25h → täglicher Import mit Matching?
- Elastizität gewünscht
  - Dynamische Anpassung der Ressourcen an den Bedarf
  - Effiziente Nutzung mehrerer Server
  - Beispiel: “25 Server für eine Stunde mieten”

Paralleles Matching in der Cloud

- Datenpartitionierung durch Blocking
  - Clustering, z.B. nach Produkttyp
- Zuordnung von Ressourcen zu Clustern
  - Unabhängiges, paralleles Matching
- Ziel: Effiziente Ausnutzung von Ressourcen
- Problem: Ungleiche Datenverteilung führt zu ungleicher Lastverteilung



Effizientes Matching in der Cloud

- Beispiel: 3 MP3-Player + 6 Mobiltelefone → 18 Paare (1 Std)

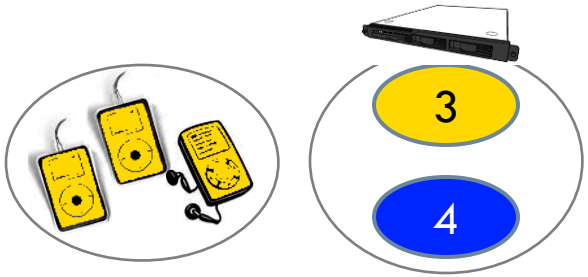
Naiver Ansatz

3 Paare
(16%)

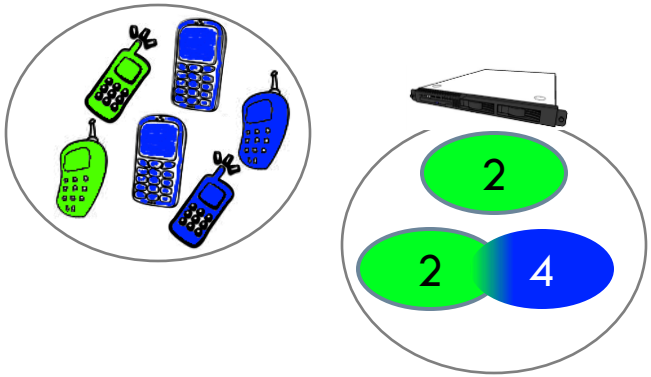
15 Paare
(84%)

Gesamt: 50min

BlockSplit-Ansatz



3 Paare
6 Paare
9 Paare (50%)



1 Paar
8 Paare
9 Paare (50%)

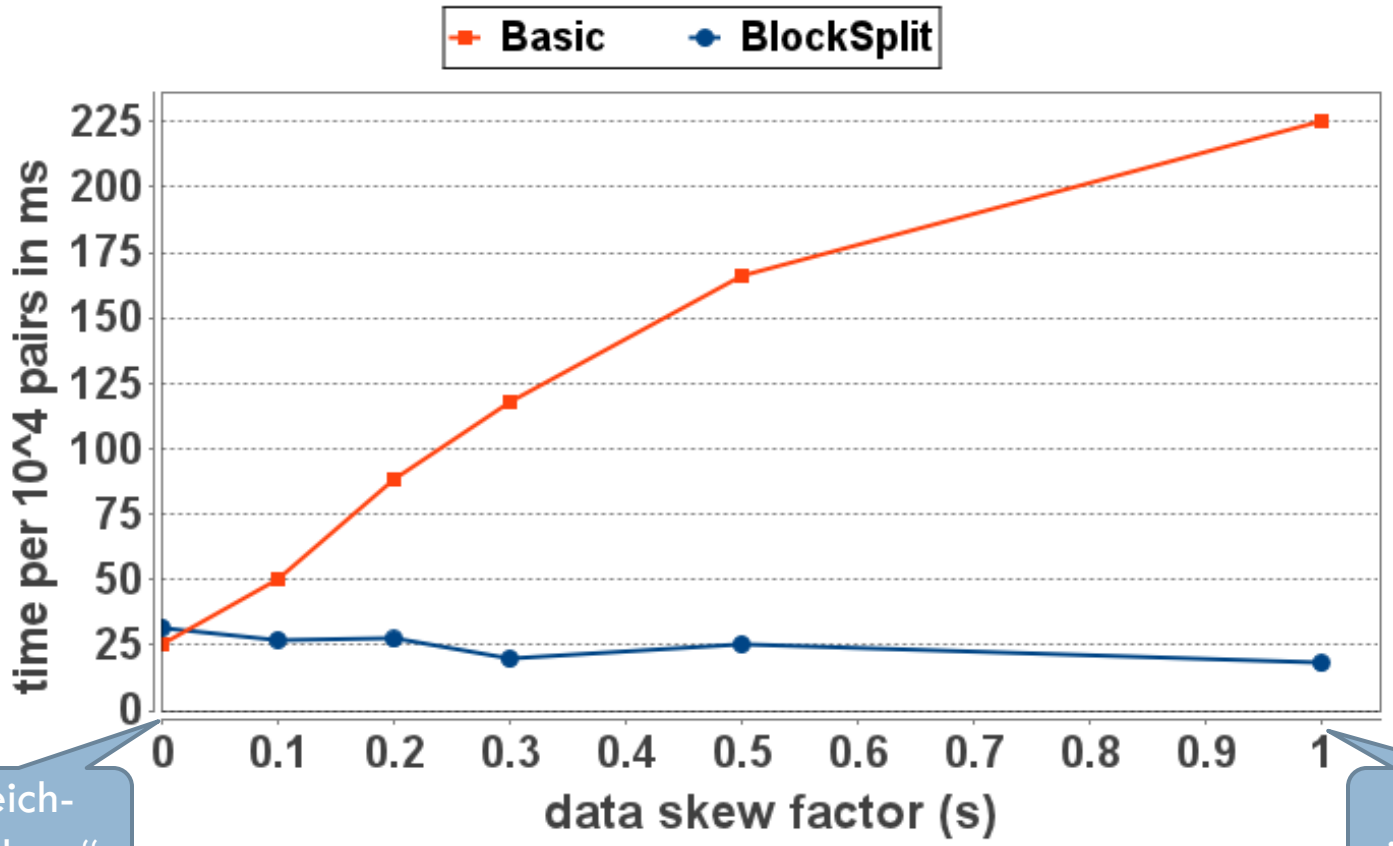
Gesamt: 30min

Realisierung von BlockSplit

- Verwendung des MapReduce-Frameworks
  - Programmierplattform für parallele Berechnungen
  - Vorteile:
    - Automatische Parallelisierung
    - Robustheit (z.B. gegenüber Server-Ausfall)
    - Skalierbarkeit (beliebig viele Server)
- Realisierung durch zwei Funktionen
  - Map: Zuordnung Produkte → Match-Task(s) → Server
  - Reduce: Matching
- Evaluierung mit Amazon EC2

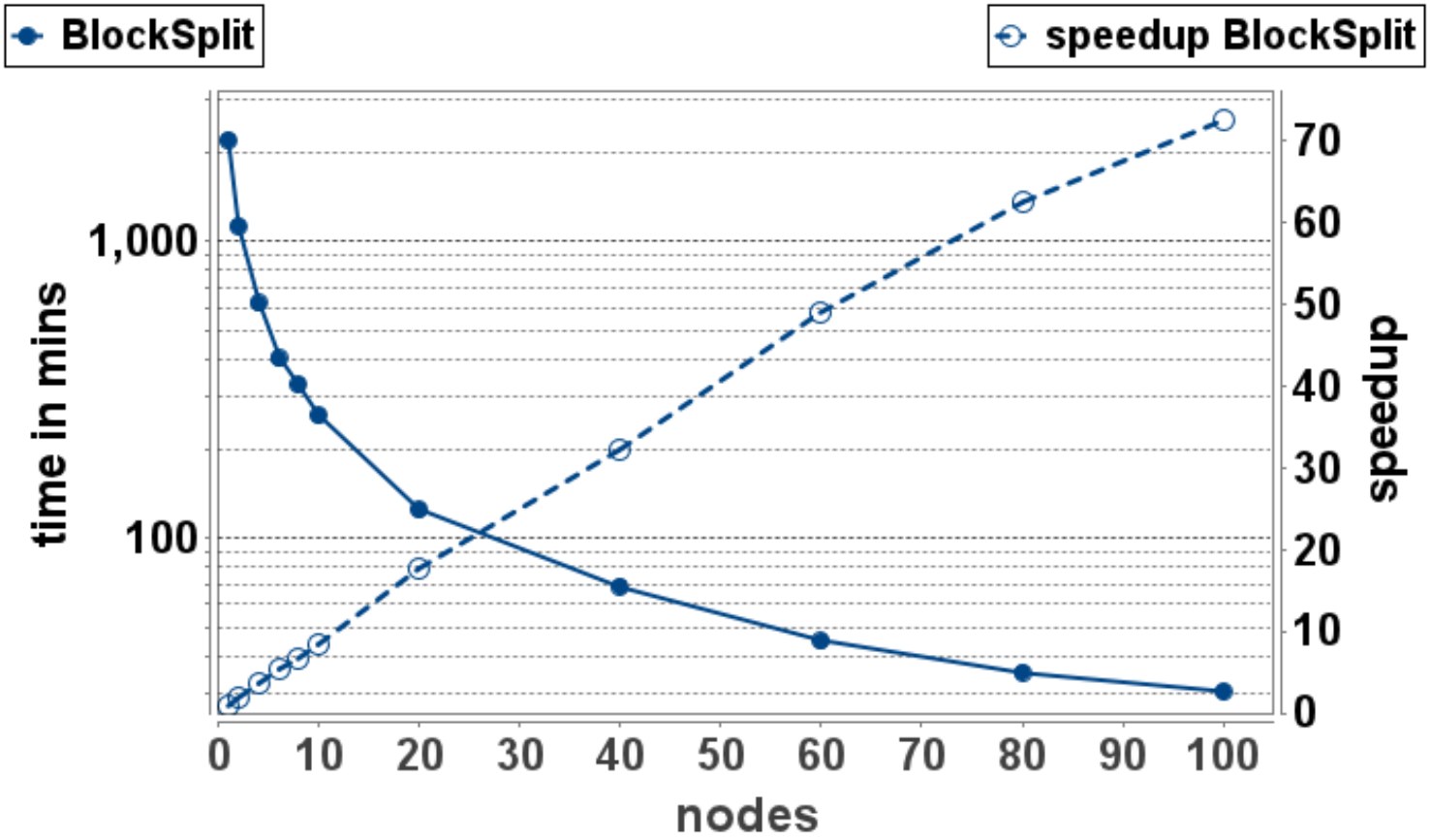
Evaluation: Robustheit

- BlockSplit-Ansatz ist robust gegenüber Datenverteilung



Evaluation: Skalierbarkeit

- BlockSplit-Ansatz ist skalierbar bzgl. Anzahl der Server



Zusammenfasung

- Cloud Computing als Mittel für ökonomisches und elastisches Produkt-Matching
- Reduzierung der Laufzeit für Produkt-Matching durch Einsatz mehrerer Server (paralleles Matching)
- Effiziente Ressourcennutzung erfordert u.a. Lastbalancierung
  - BlockSplit-Ansatz
- Weitere Forschungsthemen
  - Automatische Optimierung von Kosten (#Server) und Laufzeit
  - Andere daten- und rechenintensive Datenintegrationsaufgaben (z.B. Extraktion von Produkt-Codes)

http://dbs.uni-leipzig.de

Vielen Dank!