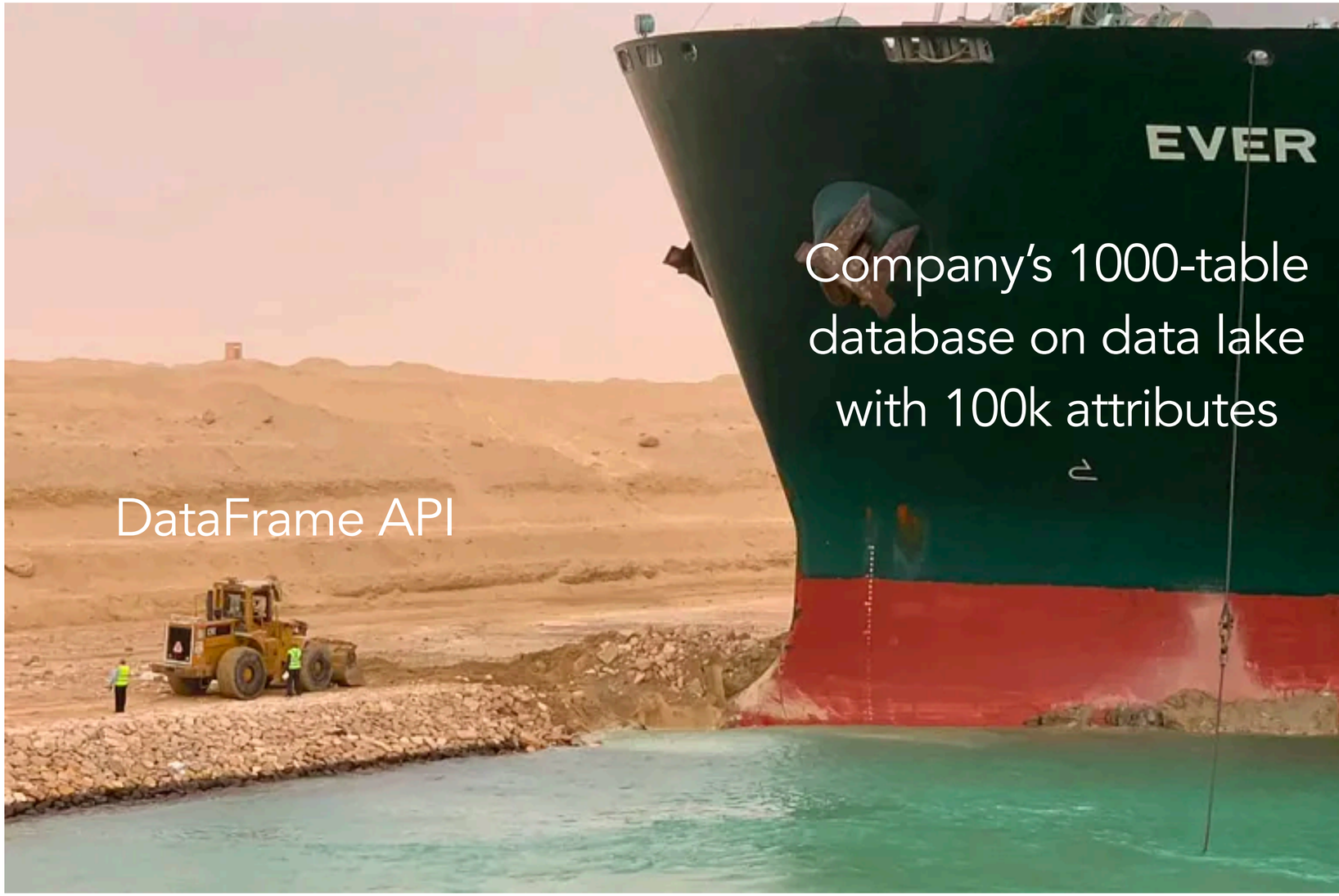
DSC 204a Scalable Data Systems

- Haojian Jin



Meme idea credit: https://datasystemsfun.tumblr.com/

Logistics

- PA0 graded.
- PA1 due date.
- Final exam.

Where are we in the class?

Foundations of Data Systems (2 weeks)

- Digital representation of Data → Computer Organization → Memory hierarchy → Process → Storage

Scaling Distributed Systems (3 weeks)

- Cloud → Network → Distributed storage → Parallelism → **Partition and replication**

Data Processing and Programming model (5 weeks)

- Data Models evolution → Data encoding evolution → → IO & Unix Pipes → Batch processing (MapReduce) → Stream processing (Spark)

Today's topic: Replication & Partitioning

- PIA
- Share-X parallelism
- Replication
- Partitioning

- Textbook:
  DDIA Ch. 5, 6

Technologies are always evolving.

A Case for Redundant Arrays of Inexpensive Disks (RAID)

David A Patterson, Garth Gibson, and Randy H Katz

Computer Science Division
Department of Electrical Engineering and Computer Sciences
571 Evans Hall
University of California
Berkeley, CA 94720
(pattrsn@ginger berkeley edu)

Increasing performance of CPUs and memories will be ıot matched by a simılar performance ıncrease ın I/O While ' Single Large Expensive Disks (SLED) has grown rapıdly, ıce ımprovement of SLED has been modest Redundant :xpensive Disks (RAID), based on the magnetic disk :veloped for personal computers, offers an attractive ;LED, promısıng ımprovements of an order of magnıtude ın reliabılity, power consumptıon, and scalabılity This paper levels of RAIDs, gıvıng their relative cost/performance, and) to an IBM 3380 and a Fujıtsu Super Eagle

1 Background: Rısing CPU and Memory Performance

The users of computers are currently enjoying unprecedented growth ın the speed of computers Gordon Bell said that between 1974 and 1984, single chip computers ımproved ın performance by 40% per year, about twice the rate of mınicomputers [Bell 84] In the following year Bill Joy

ure of magnetic disk technology is the growth in the maximum num bits that can be stored per square inch, or the bits per inch in a times the number of tracks per inch Called M A D , for maxima density, the "First Law in Disk Density" predicts [Frank87]

$$MAD = 10^{(Year-1971)/10}$$

Magnetic disk technology has doubled capacity and halved price ever years, in line with the growth rate of semiconductor memory, a practice between 1967 and 1979 the disk capacity of the average IB processing system more than kept up with its main memory [Steve

Capacity is not the only memory characteristic that must rapidly to maintain system balance, since the speed with instructions and data are delivered to a CPU also determines its u performance The speed of main memory has kept pace for two reas (1) the invention of caches, showing that a small buffer can be ma

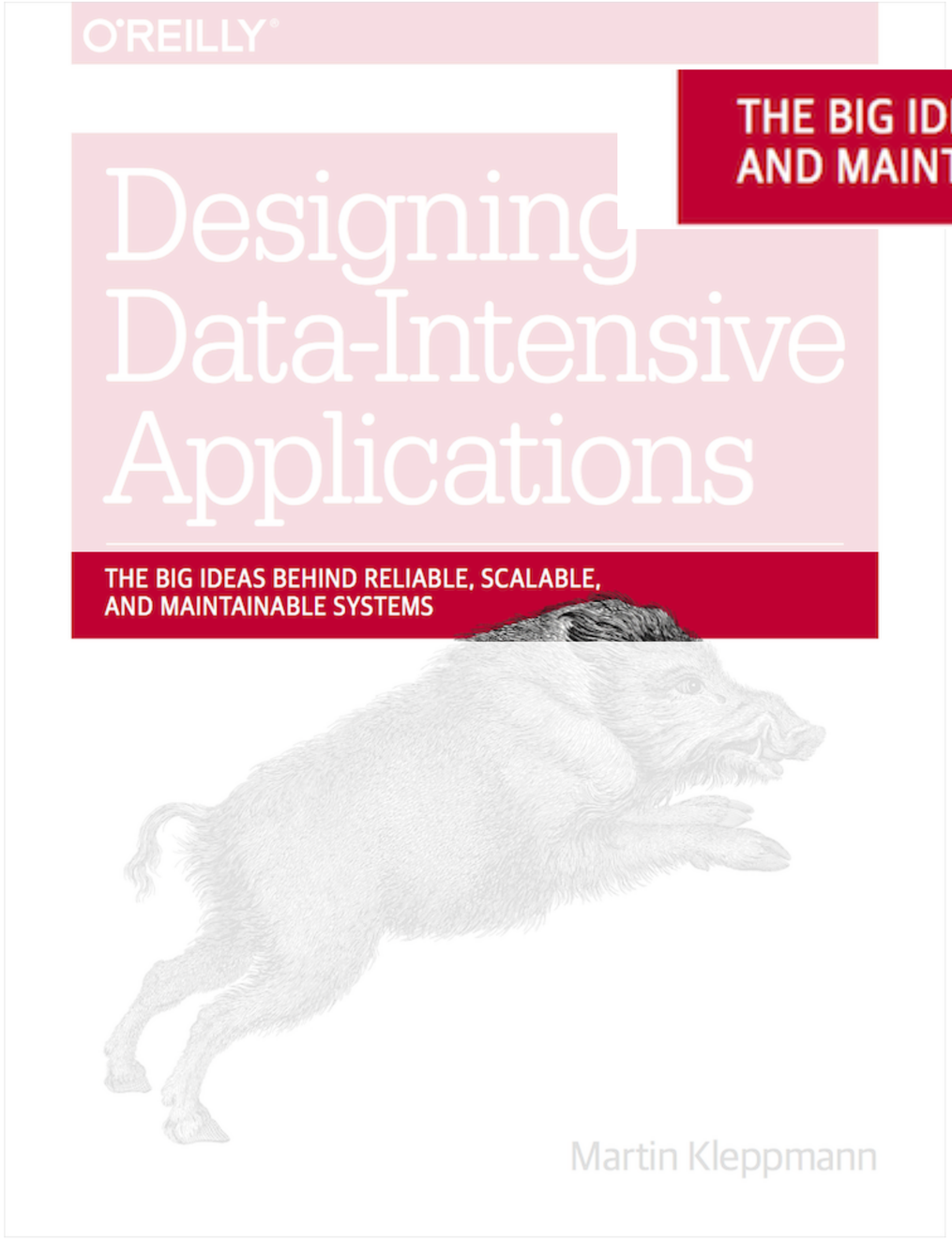
Technologies are always evolving.

Market Trends—USB Flash Drives, Worldwide, 2001–2010

Average Selling Price of USB Flash Drives, Worldwide, 2001-2010 (Dollars)—Actual and Projected

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	CAGR 2004-2010
8MB	11.13	8.20	9.75	-	-	-	-	-	-	-	NA
16MB	16.14	11.37	9.51	6.85	6.18	-	-	-	-	-	-100%
32MB	25.68	17.41	12.22	8.04	6.36	5.54	-	-	-	-	-100%
64MB	41.53	27.69	19.52	14.01	8.08	6.37	5.08	4.34	-	-	-100%
128MB	78.18	50.16	29.95	22.16	12.05	8.21	6.01	5.50	4.58	4.07	-25%
256MB	156.83	98.50	51.11	33.10	19.30	12.44	7.69	6.72	5.94	4.92	-27%
512MB	326.34	196.13	112.77	57.79	31.04	20.17	11.58	8.92	7.36	6.69	-30%
1GB	-	410.74	214.31	88.48	55.91	32.69	18.66	14.00	9.95	8.56	-32%
2GB	-	-	3,335.43	166.56	102.86	59.19	30.14	23.28	15.91	11.94	-36%
4GB	-	-	-	-	224.41	109.25	54.44	38.30	26.79	19.72	NA
8GB	-	-	-	-	555.88	238.81	100.34	70.11	44.40	33.92	NA
16GB	-	-	-	-	-	592.17	219.14	130.19	81.70	56.92	NA
32GB	-	-	-	-	-	1,416.67	543.15	285.70	152.16	105.63	NA
64GB	-	-	-	-	-	-	1,212.76	709.82	334.51	197.64	NA
Average	**21.09**	**31.22**	**29.10**	**28.40**	**28.48**	**25.89**	**20.94**	**21.15**	**19.77**	**20.71**	**-5%**

Source: Joseph Unsworth, Gartner Dataquest, August 2005 (used with permission).



THE BIG IDEAS BEHIND RELIABLE, SCALABLE, AND MAINTAINABLE SYSTEMS

- No silver bullets
  - Lots of trade-off
- Constantly evolving
  - New proposals
    - Succeed → New standard/product
    - Failed → Ideas incorporated
- No black magic
  - Keep revisiting old BIG ideas
- Ask me questions!

Redundancy

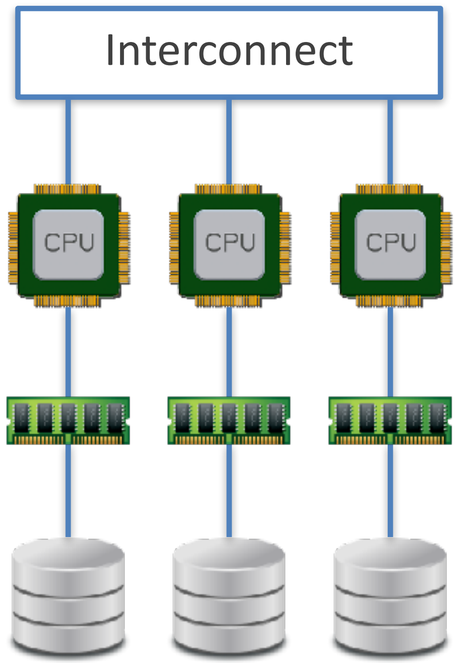
- How many computers are there in the space shuttle?
  - Five general-purpose computers.
    - Four operate during critical mission phase (e.g., ascent, decent)
    - One backup.
  - Four actuators.

Computers in the Space Shuttle Avionics System
https://history.nasa.gov/computers/Ch4-4.html

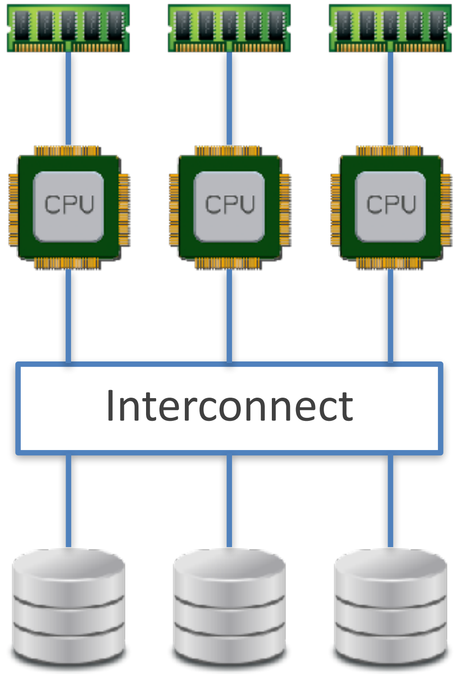
Why distribute a database across multiple machines?

- Scalability
  - Data volume
  - Read load
  - Write load
- Fault tolerance | high availability
  - When one fails, another can take over.
- Latency
  - Distribute machines worldwide.
  - Reduce network latency.

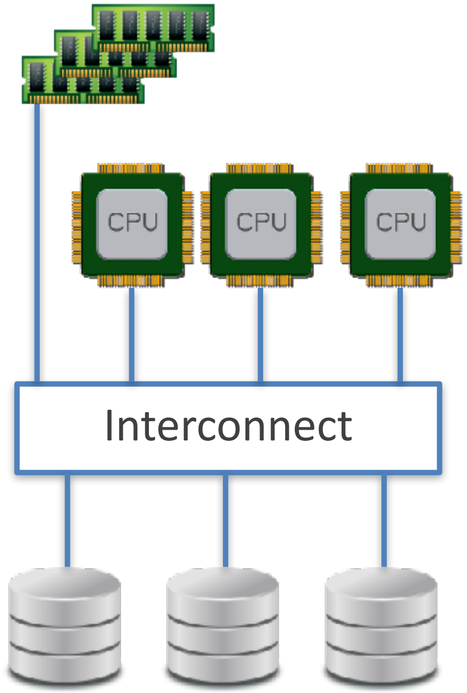
3 Paradigms of Multi-Node Parallelism



Shared-Nothing Parallelism



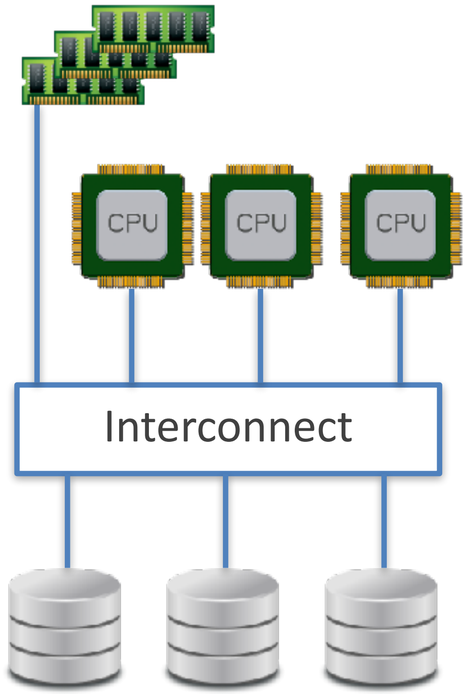
Shared-Disk Parallelism



Shared-Memory Parallelism

Shared-memory parallelism (vertical scaling)

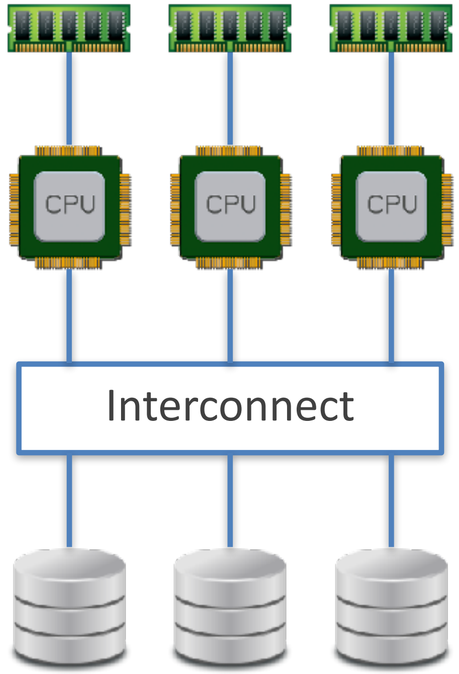
- CPUs, RAM chips, Disks are joined under one OS.
- Advantage:
  - Simple, High performance.
- Disadvantage:
  - Expensive.
    - Cost grows faster super-linearly.
    - Performance grows sub-linearly.
  - Limited fault tolerance
    - Note hot-swappable
  - Geolocation
- App: Large language model training.



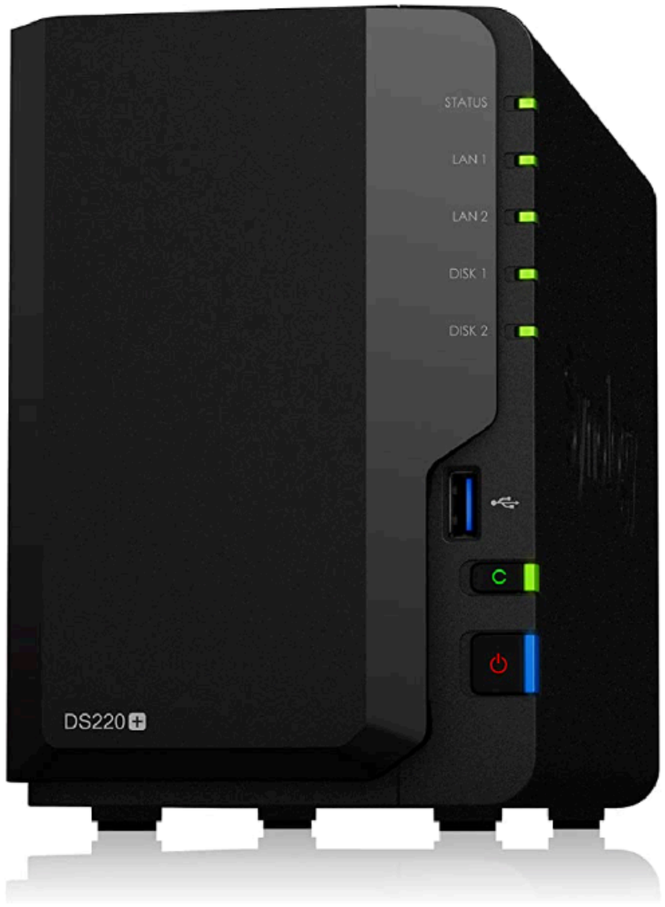
Shared-Memory Parallelism

Shared-Disk Parallelism (data warehouse)

- Machines with independent CPUs and RAM
  - But stores data on an array of disks
- Advantage:
  - Low-cost
- Disadvantage:
  - Overhead of locking limit the scalability



Synology 2 bay NAS DiskStation DS220+ (Diskless),Black



Roll over image to zoom in

Brand: Synology
4.7 ★★★★½ 3,538 ratings | 322 answered questions
#1 Best Seller in Network Attached Storage (NAS) Enclosures

$299.99

FREE Returns

Pay $25.00/month for 12 months, interest-free upon approval for Amazon Visa

Available at a lower price from other sellers that may not offer free Prime shipping.

Extra Savings 90 days FREE music unli… 1 Applicable Promotion

Size: **2-bay; 2GB DDR4**

2-bay; 2GB DDR4 | 4-bay; 2GB DDR4 | 4-bay; 4GB DDR4

Style: **DS220+**

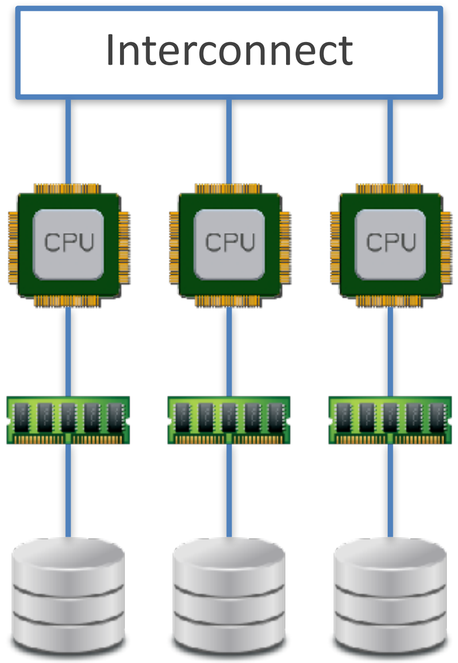
DS220+ | DS723+ | DS723+ + D4ES01-16G | DS723+ + E10G22-T1-Mini | DS723+ + SNV3410-400G | DS723+ + SSD SNV3410 800GB | DS423+ | DS923+

Brand	Synology
Color	Black
Item Dimensions LxWxH	6.5 x 4.25 x 9.14 inches
Size	2-bay; 2GB DDR4
Compatible Devices	Smartphone,Television,Laptop,Desktop,Camera

About this item

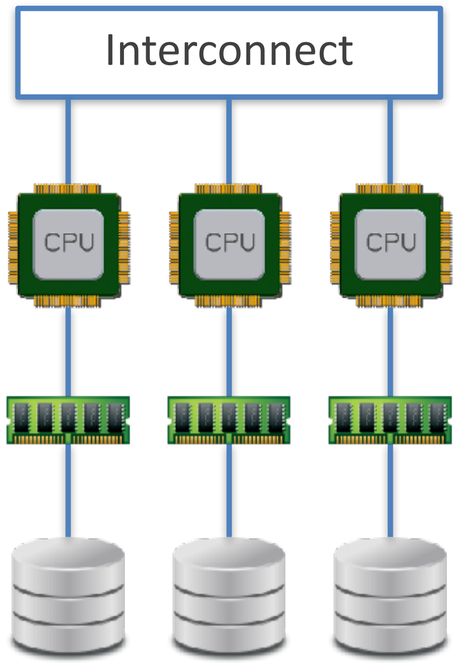
- Featured dual 1GbE LAN ports to support network failover, and with Link Aggregation enabled, DS220+ provides over 225 MB/s sequential read and 192 MB/s sequential write throughput. Data can be further protected with RAID 1 disk mirroring to prevent sudden drive failure.
- Intel dual-core processor with AES-NI hardware encryption engine; 2 GB DDR4 memory (expandable up to 6 GB)

Shared-Nothing Parallelism (horizontal scaling)



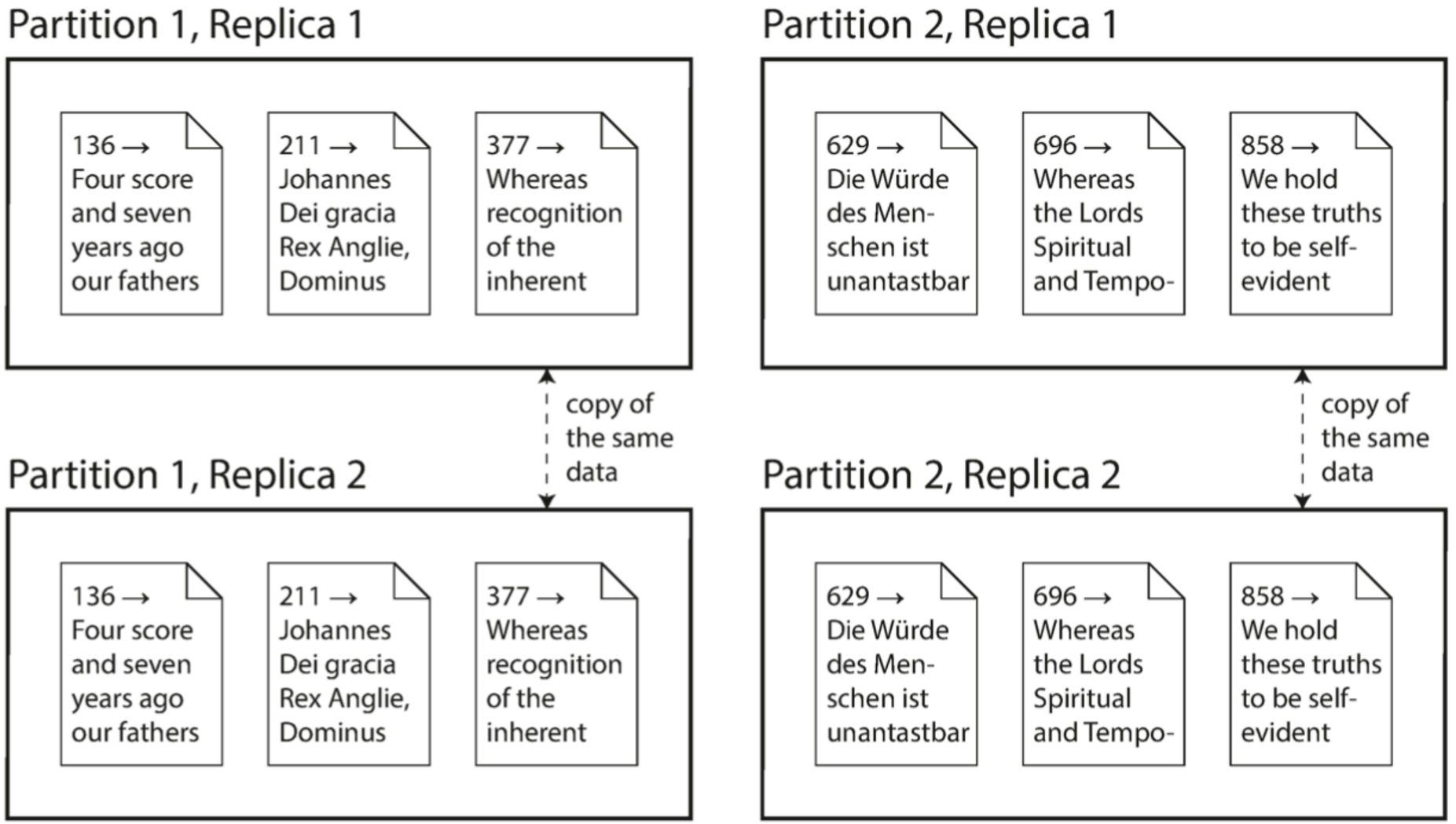
- Most popular approach!
- Each node uses its CPUs, RAM, and disks independently.
  - Any coordination, software level, through a conventional network.
- No silver bullet.
  - Not always the best one.

Shared-Nothing Parallelism (horizontal scaling)



- Advantage:
  - Performance
  - Cost
- Disadvantage
  - Complexity.
  - Involves many constraints and trade-offs.
- Database cannot magically hide all these from you.

Replication versus Partitioning



Today's topic: Replication & Partitioning

- PIA
- Share-X parallelism
- **Replication**
- Partitioning

- Textbook:
  DDIA Ch. 5, 6

Recall:

Why distribute a database across multiple machines?

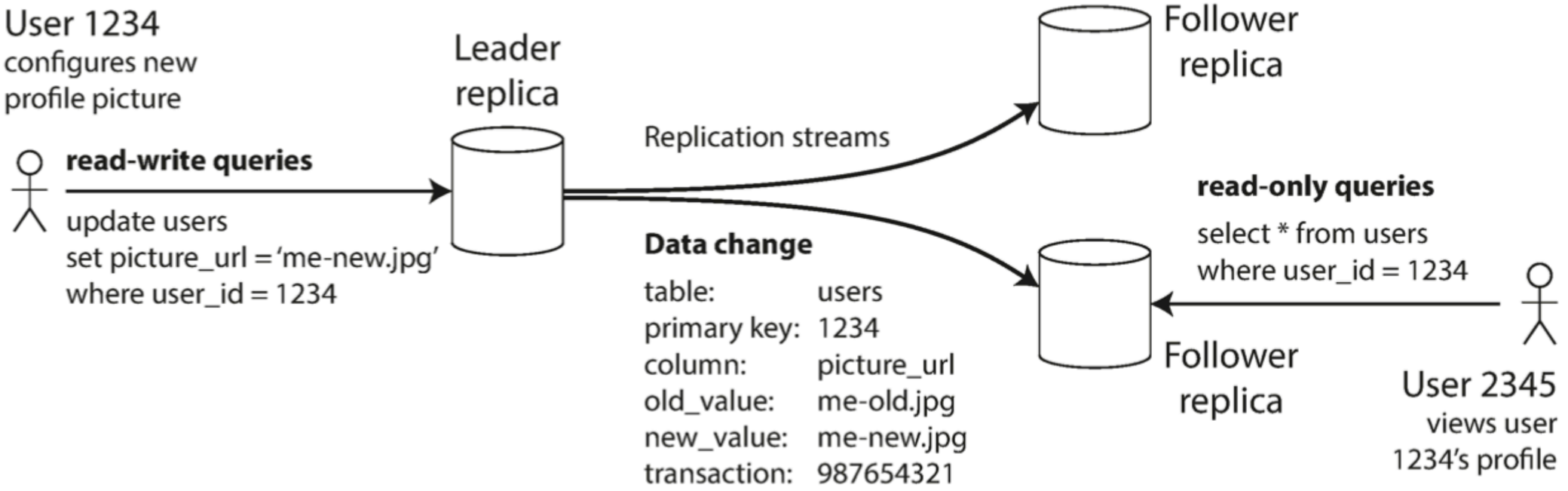
- Scalability
  - Data volume
  - Read load
  - Write load
- Fault tolerance | high availability
  - When one fails, another can take over.
- Latency
  - Distribute machines worldwide.
  - Reduce network latency.

Replication (assuming a small dataset)

- Scalability
  - Data volume
  - Read load
  - Write load
- Fault tolerance | high availability
  - When one fails, another can take over.
- Latency
  - Distribute machines worldwide.
  - Reduce network latency.

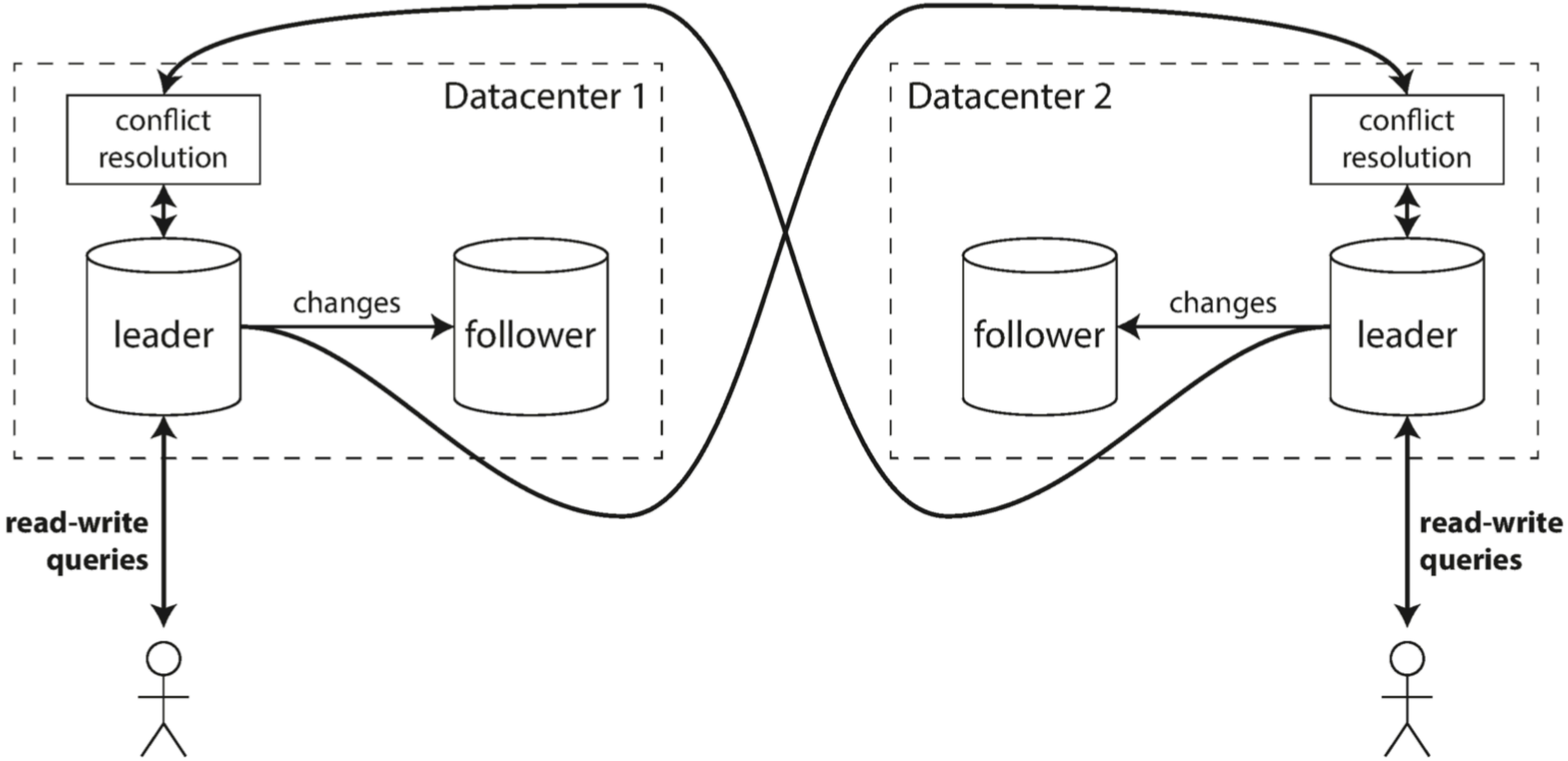
Challenge: How to handle changes to replicated data?

Leaders and Followers



- Clients send requests to the leader to write to the db. The leader saves locally
- The leader then sends the data change to all of its followers.
- Clients read data from the leader or followers.
- One of the big ideas in CS (master-slave). Distributed msg brokers (e.g., Kafka).

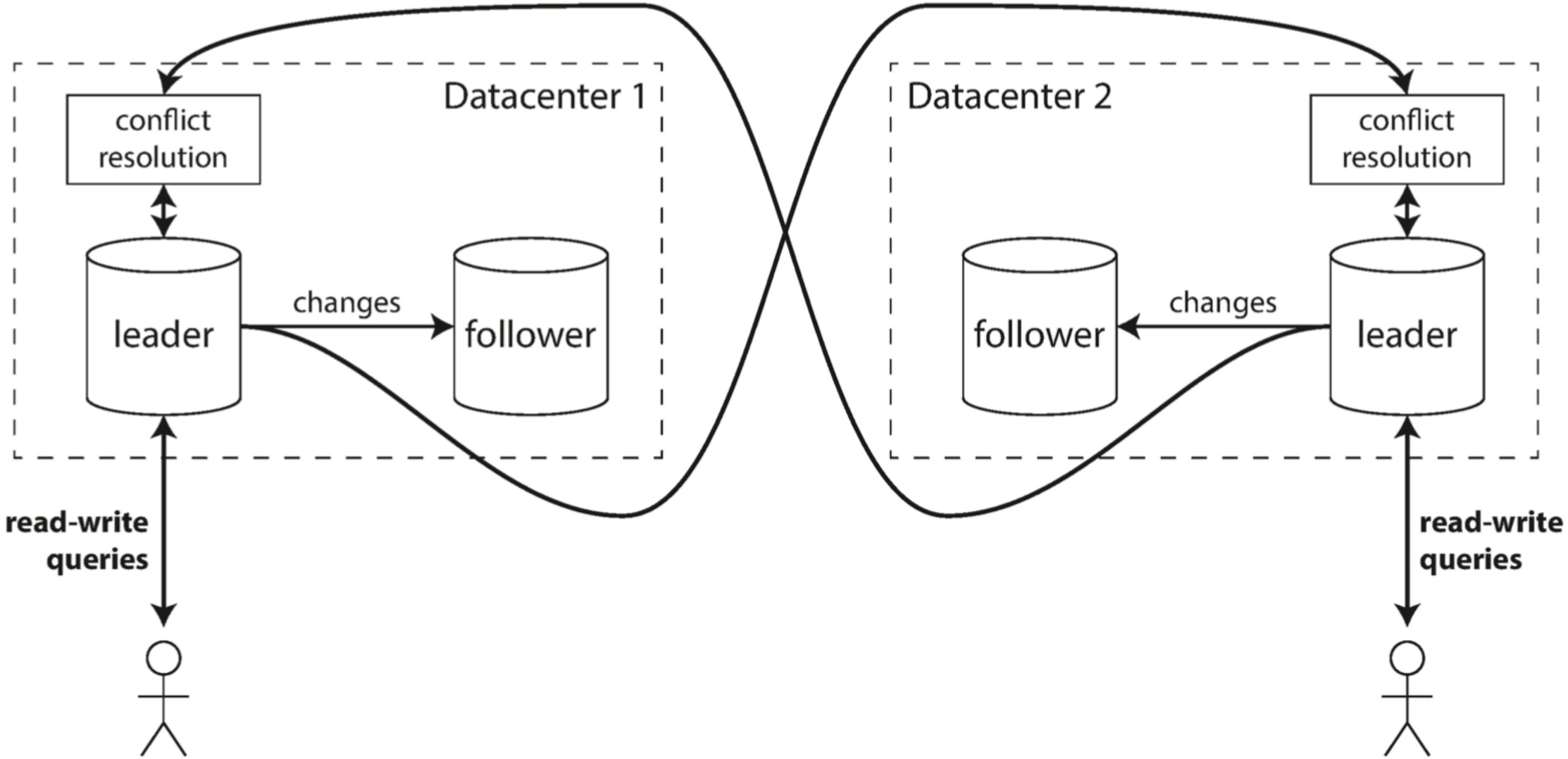
Single-leader replication



Single-leader replication

- Advantages:
  - Simplicity (easy-to-understand)
  - Conflicts across nodes (consistency)
- Disadvantages:
  - Single point of failure

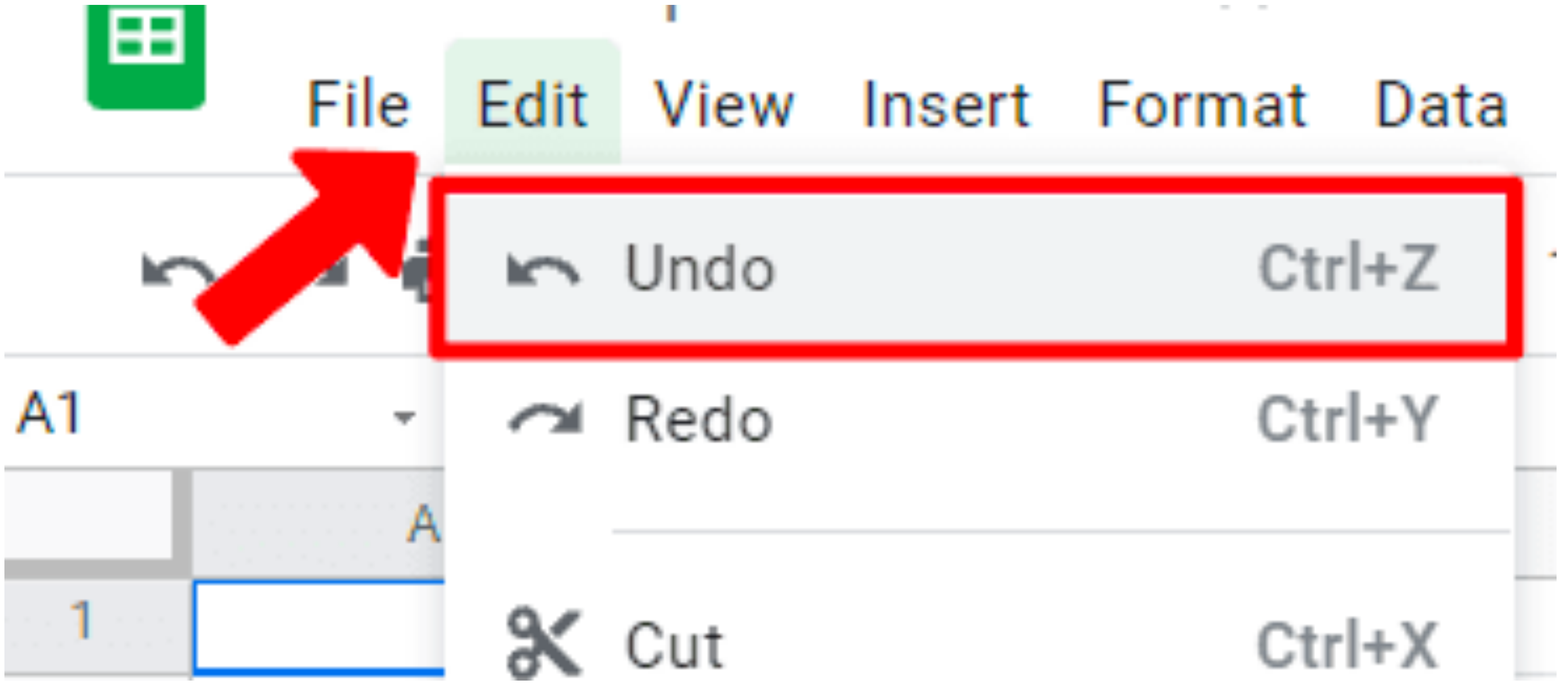
Multi-leader replication (Multi-datacenter operation)



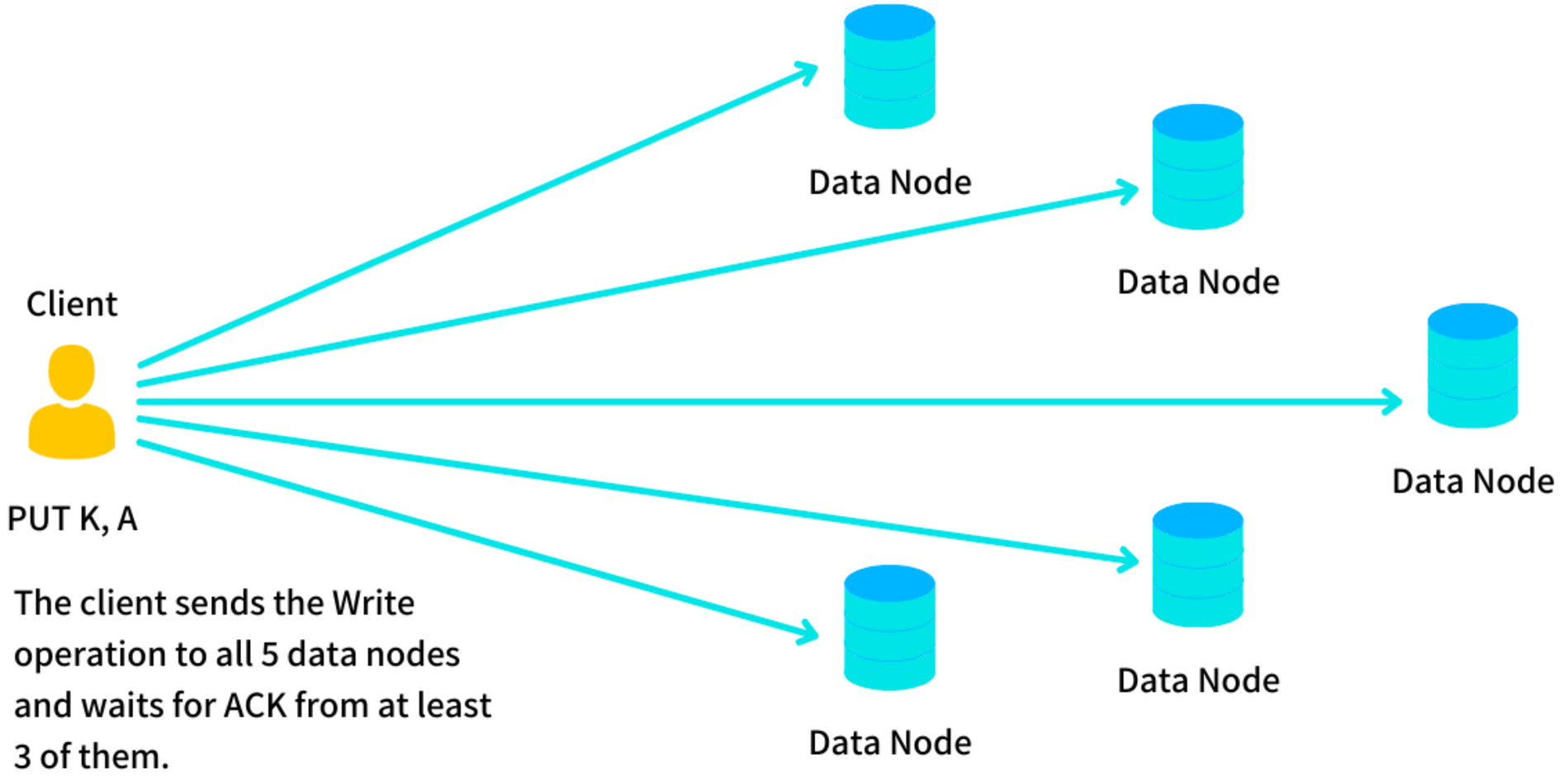
Multi-leader replication

- Advantages:
  - Performance (i.e., network latency).
  - Tolerance of data center outages.
  - Tolerance of network problems.
    - Asynchronous syncing.
- Disadvantages:
  - Potential write conflicts.
    - The same data may be concurrently modified in two different data centers.
- Only use it if necessary.

Google Doc Undo-Redo



Leaderless replication



Challenge: How to handle changes to replicated data?

- Three main approaches:
  - Single-leader replication
  - Multi-leader replication
  - Leaderless replication
- Tradeoffs
  - Simplicity (easy-to-understand)
  - Conflicts across nodes (consistency)
  - Faulty nodes
  - Network interruption
  - Latency spikes

Replication (assuming a small dataset)

- Scalability
  - Data volume
  - Read load
  - Write load
- Fault tolerance | high availability
  - When one fails, another can take over.
- Latency
  - Distribute machines worldwide.
  - Reduce network latency.

Replication (assuming a small dataset)

- Scalability
  - Data volume
  - Read load
  - Write load
- Fault tolerance | high availability
  - When one fails, another can take over.
- Latency
  - Distribute machines worldwide.
  - Reduce network latency.

Partitioning (what if the dataset is too big for a single machine?)

Bigtable: A distributed storage system for structured data [PDF] acm.org

F Chang, J Dean, S Ghemawat, WC Hsieh… - ACM Transactions on …, 2008 - dl.acm.org

… Despite these varied demands, **Bigtable** has … **Bigtable**, which gives clients dynamic control over data layout and format, and we describe the design and implementation of **Bigtable**…

☆ Save 99 Cite Cited by 7813 Related articles All 229 versions

[PDF] The **Google file system** [PDF] acm.org

S Ghemawat, H Gobioff, ST Leung - … symposium on Operating **systems** …, 2003 - dl.acm.org

… the **Google File** System, a scalable distributed **file system** for … goals as previous distributed **file systems**, our design has … departure from some earlier **file system** assumptions. This has led …

☆ Save 99 Cite Cited by 10064 Related articles All 305 versions

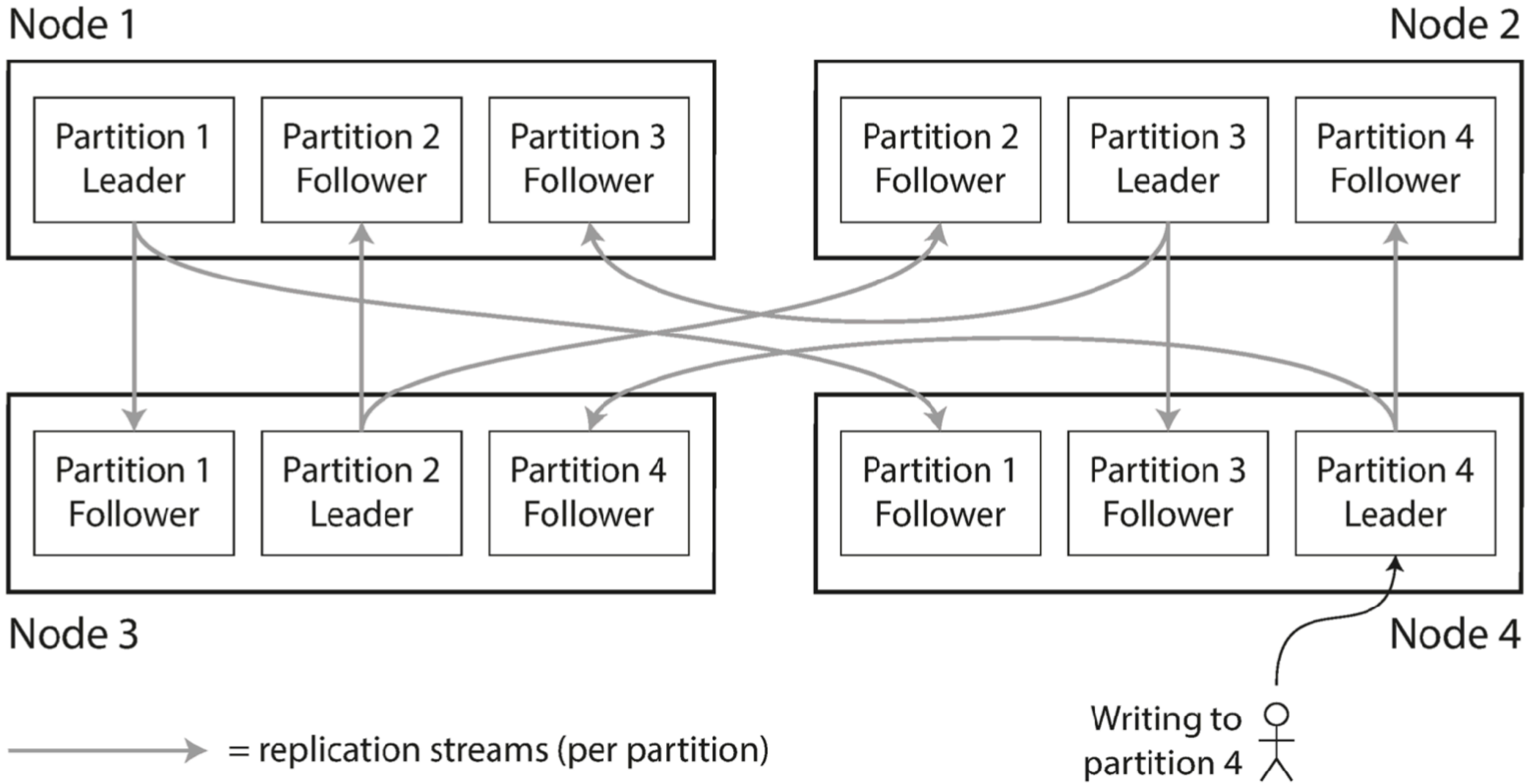
MapReduce: simplified data processing on large clusters [PDF] acm.org

J Dean, S Ghemawat - Communications of the ACM, 2008 - dl.acm.org

… pairs, and then applying a **reduce** operation to all the values … with user-specified **map** and **reduce** operations allows us to … implementation of the **Map Reduce** interface tailored towards …

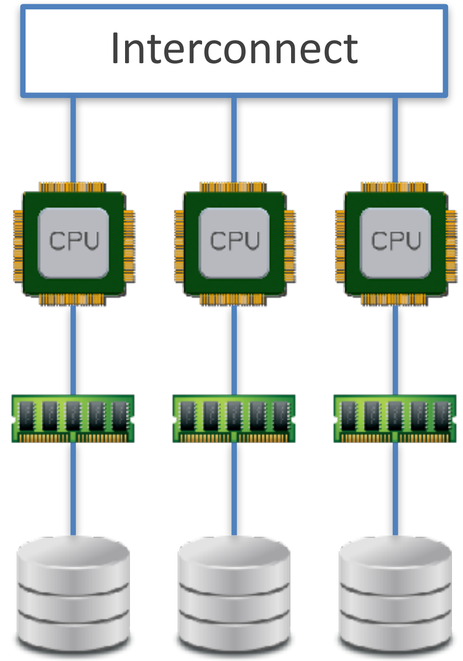
☆ Save 99 Cite Cited by 22507 Related articles All 96 versions

Combining replication and partitioning



Intuitions behind partitioning (why?)

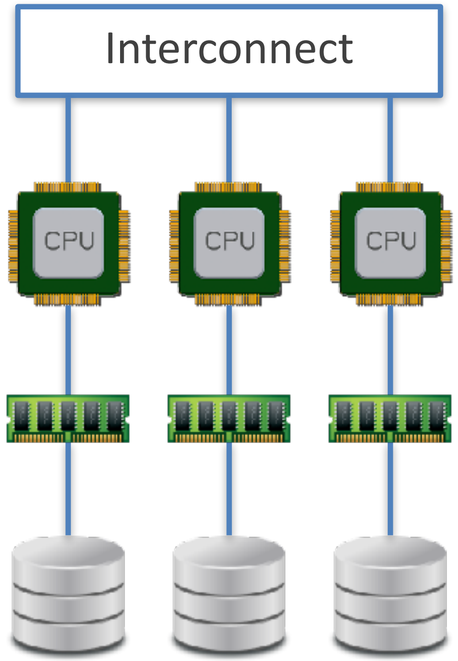
- A large dataset can be distributed across many disks.
- The query load can be distributed across many processors.



Shared-Nothing Parallelism

Partitioning challenges

- How to partition and how to index?
- How to add or remove nodes?
- How to route the requests and execute queries?

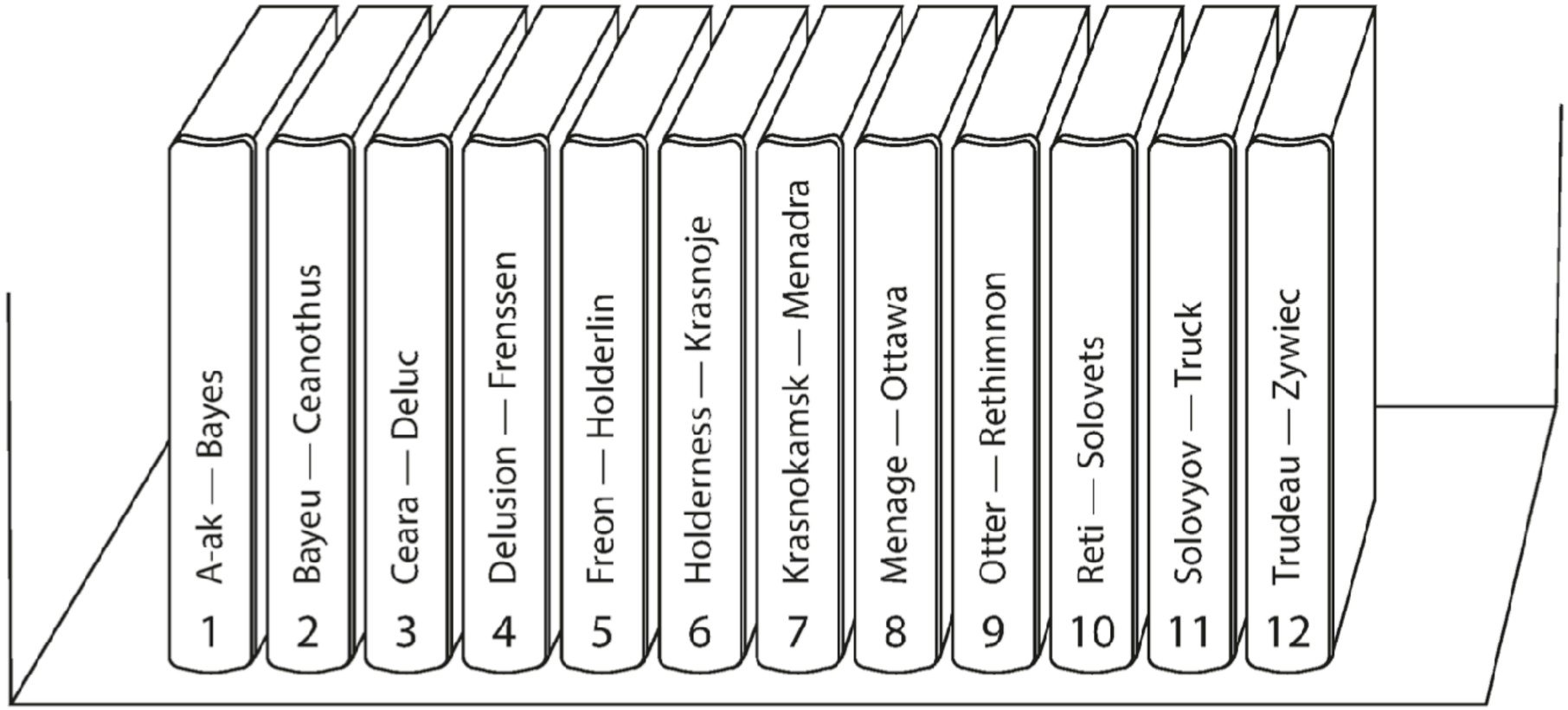


Shared-Nothing Parallelism

How to partition?

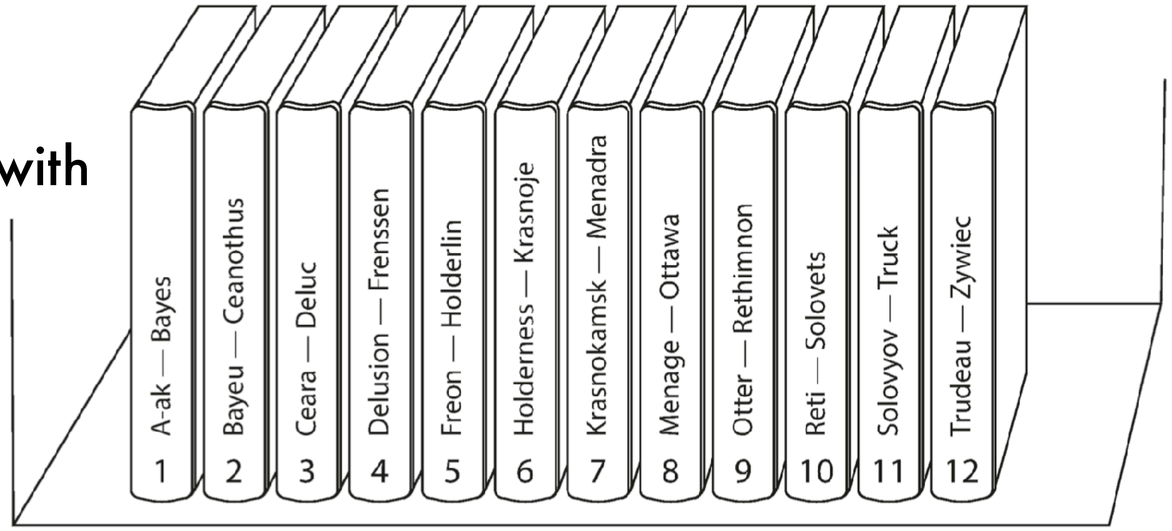
- Ideally:
  - Spread the data and the query load evenly across nodes.
- Reality:
  - Hot spot: a partition with disproportionately high load.
- Straw-man solution:
  - Distribute randomly.
  - Problem:
    - When you read a data, you do not know which node you should request. You have to request all the nodes.

Partition solution #1: by Key Range

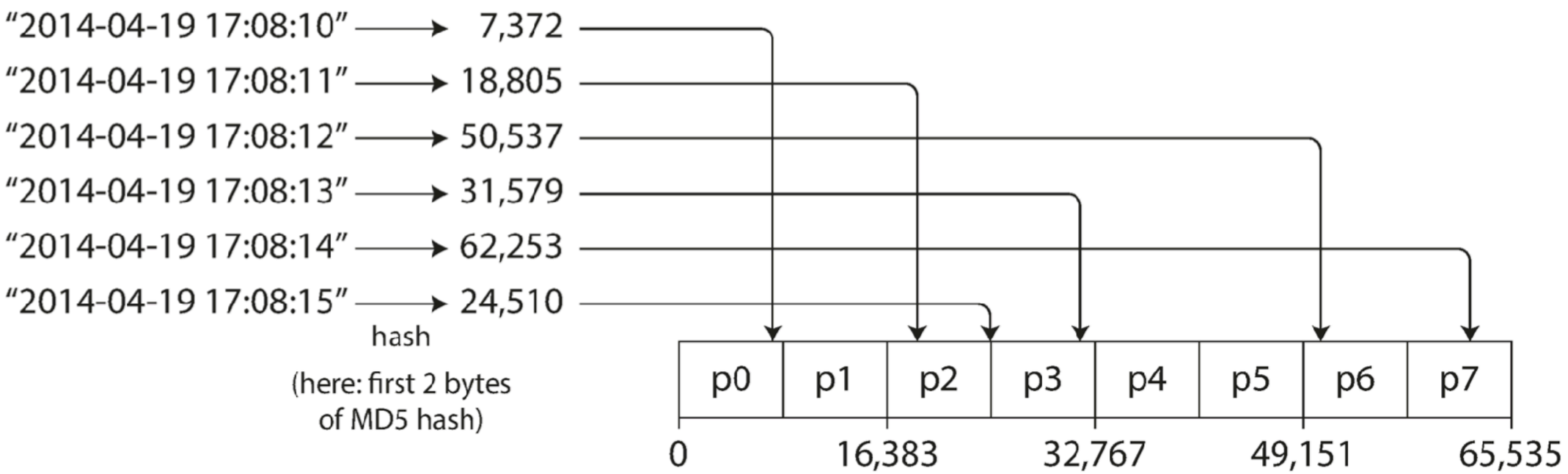


Partition solution #1: by Key Range

- Advantage: can do range queries
- Problems:
  - The ranges of keys are not necessarily evenly spaced.
    - Volume 12 contains words starting with T - Z.
  - Manual process. Require domain expertise.
  - Hard to rebalance.
  - Hot spot issues.
    - One letter is popular.
- Common keys: names, titles, dates.



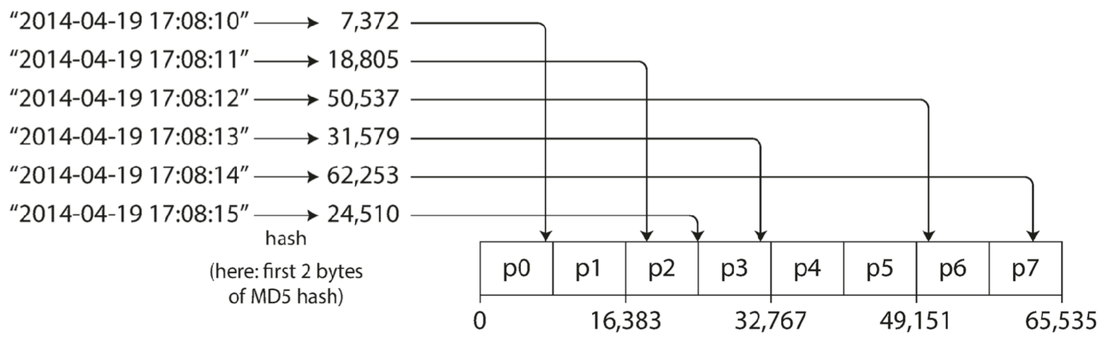
Partition solution #2: by Hash of Key



Recall bloom filter.

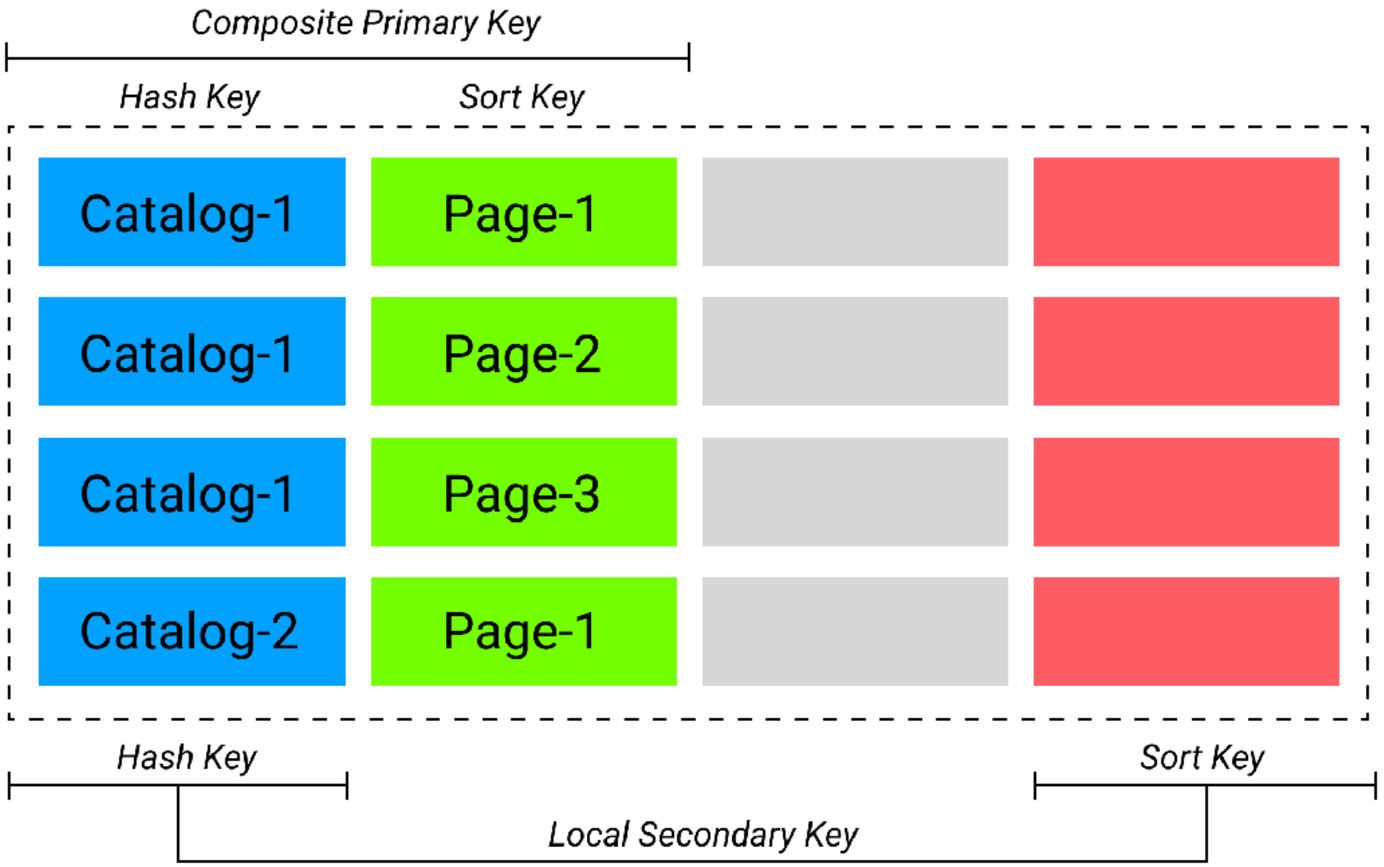
Partition solution #2: by Hash of Key

- Advantages:
  - Automatic.
  - Easy to balance.
- Problems:
  - Do not support efficient range queries.



Recall bloom filter.

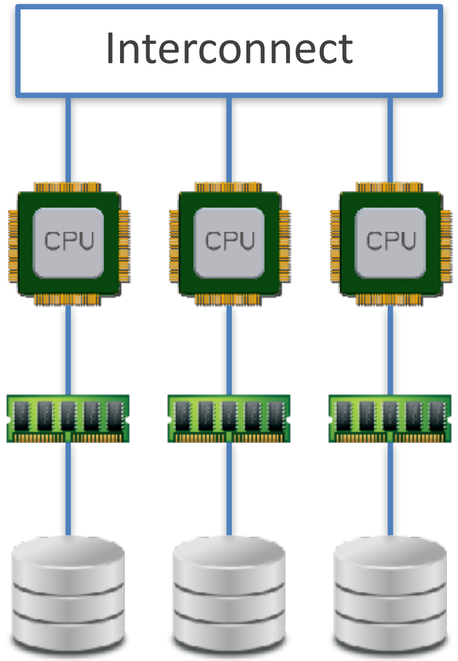
Partition solution #3: Hash of Key + Key Range



user_id update_timestamp

Partitioning challenges

- How to partition and how to index?
- **How to add or remove nodes?**
- How to route the requests and execute queries?

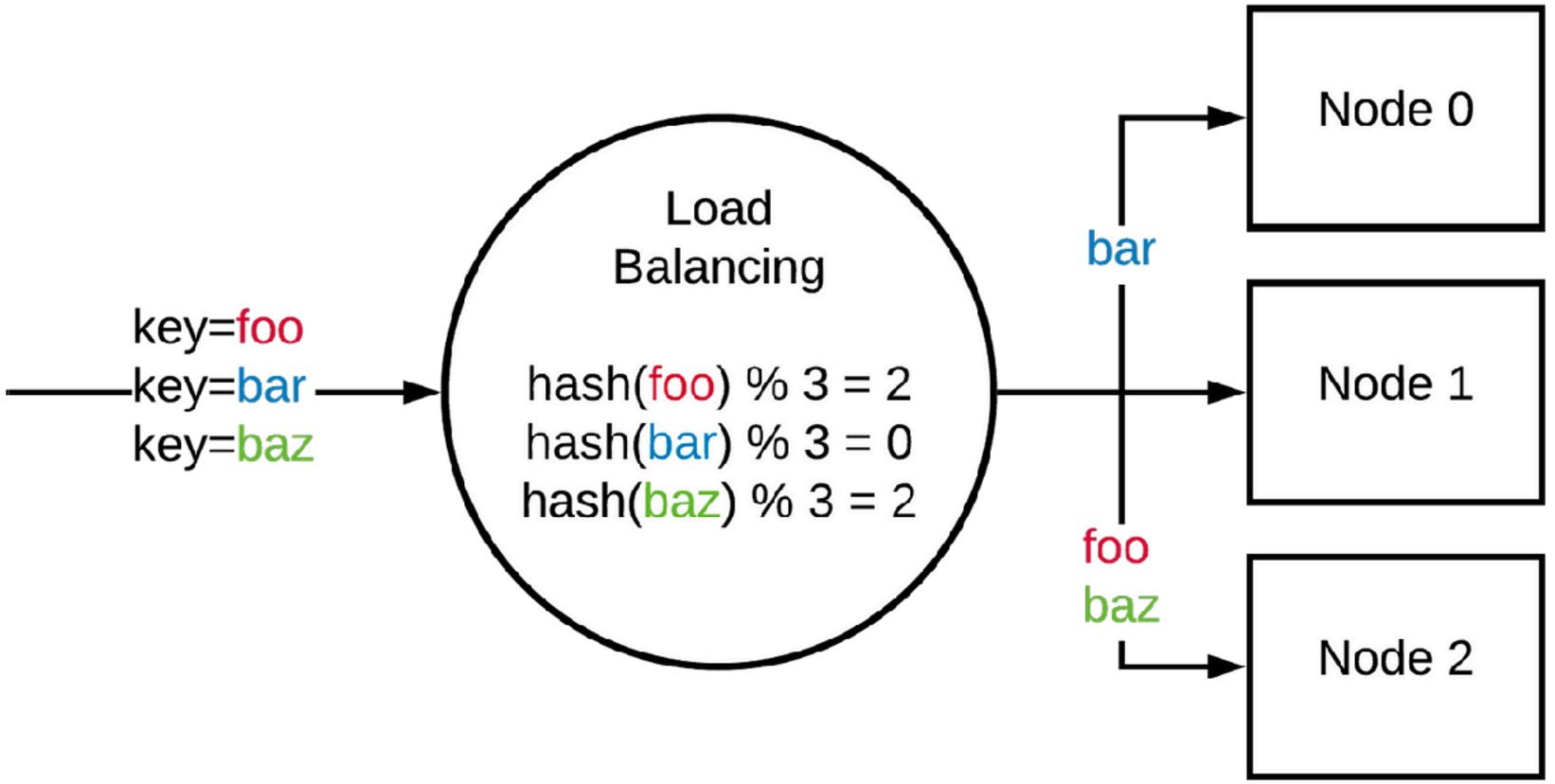


Shared-Nothing Parallelism

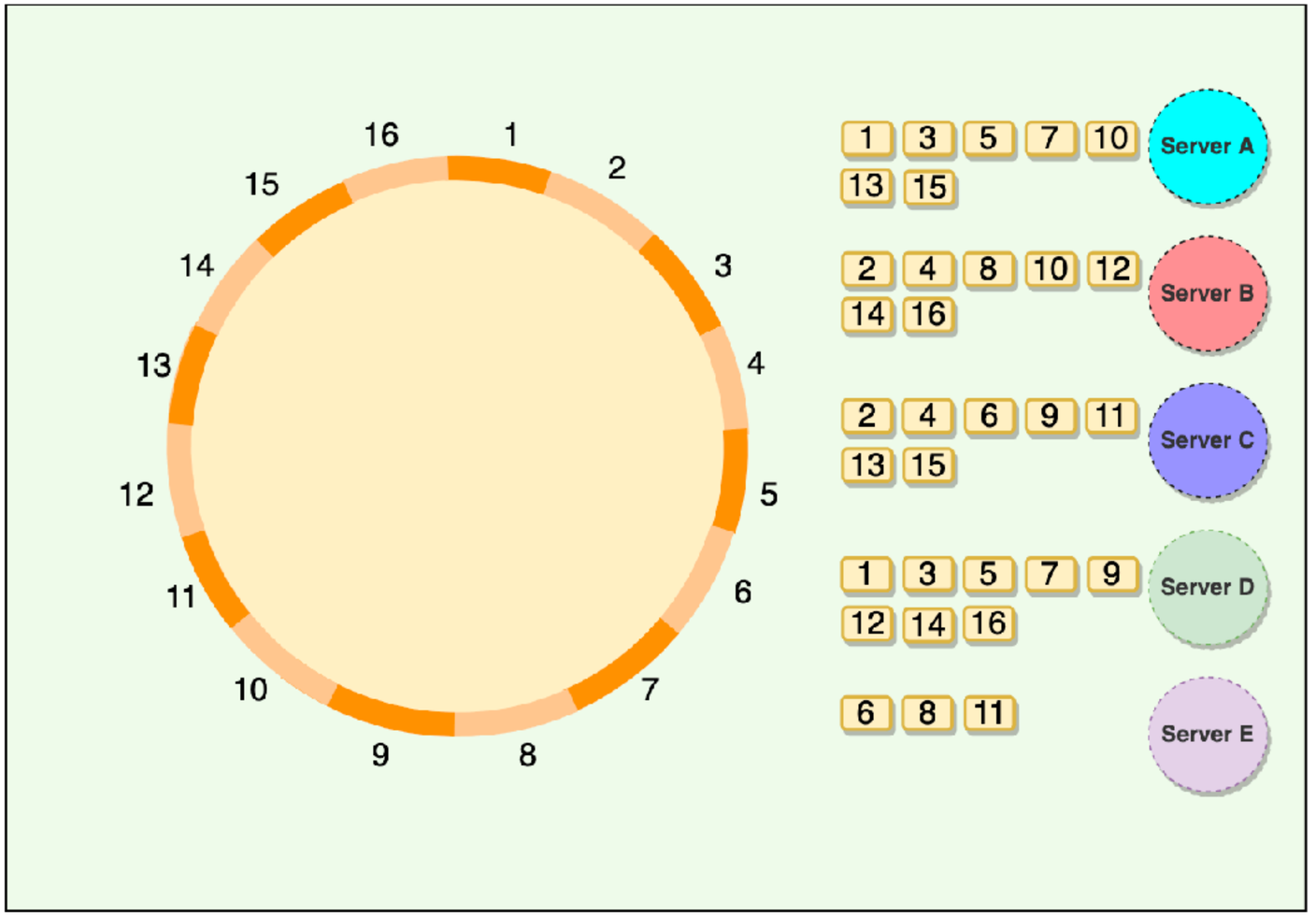
Rebalance

- Move the load from one node in a cluster to another
  - The query throughput increases → more CPUs
  - The dataset size increases → more disks and RAM
  - Machine failure
- Rebalancing goals
  - Share the load fairly after rebalancing.
  - Continue accepting reads and writes while rebalancing.
  - Minimize data moving (i.e., network and disk I/O load)

Strawman solution: Hash mod N



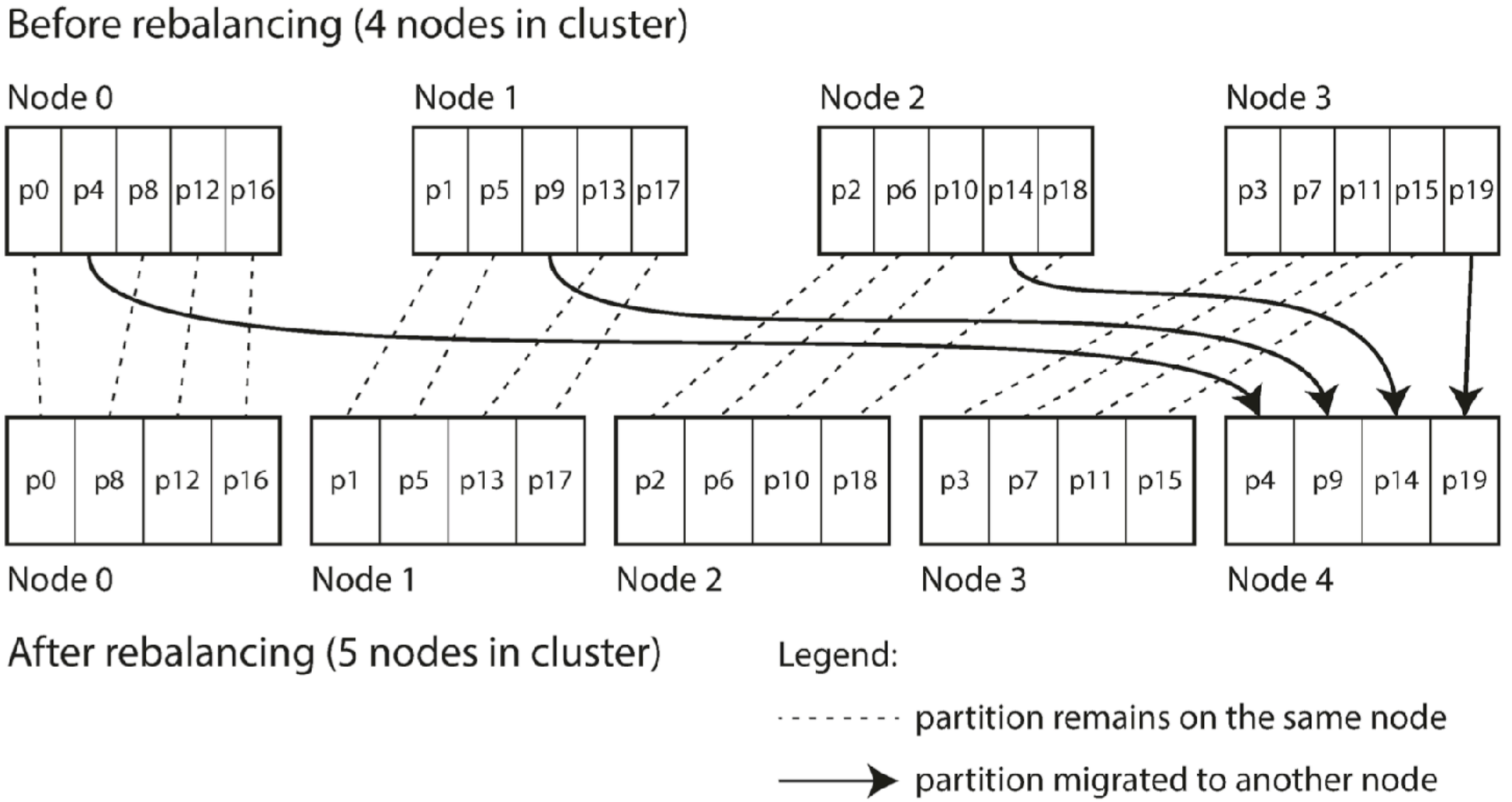
Consistent hashing ring



Mapping Vnodes to physical nodes on a Consistent Hashing ring

Learn more in https://www.youtube.com/watch?v=UF9Iqmg94tk

Rebalancing solution #1: Fixed number of partitions



Rebalancing solution #1: Fixed number of partitions

- The total number of partitions is fixed.
- The # of nodes can be adaptive for different machines.
- Partitions should have similar sizes => why?
  - Easier for management.
- Each partition grows proportionally to the total amount of data.
- Challenges:
  - Need to choose the right number of partitions.
    - Too high → too much overhead
    - Too low → Migration will be very expensive.
  - Wrong index system.
    - Very unbalanced distributions.
  - Only work with hash partitioned database. => Why?

Rebalancing solution #2: Dynamic partitions

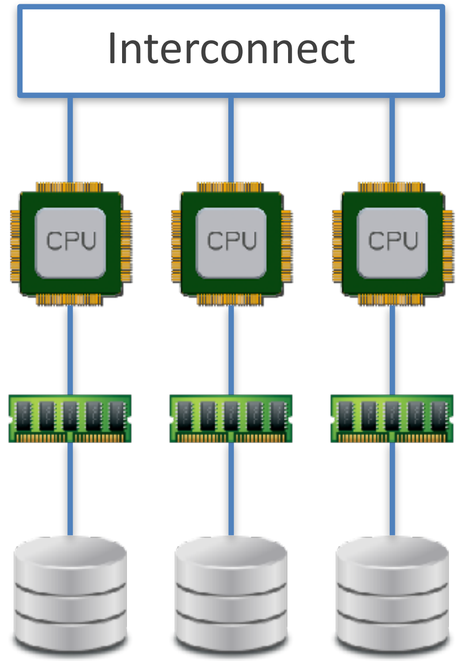
- Similar idea as the B-tree.
- Split ← When a partition grows to exceed a configured size (e.g., 10 GB).
- Merge ← When lots of data is deleted and a partition shrinks.
- One node → multiple partitions.
- One partition → one node.
- A caveat:
  - By default, start → an empty database → one partition → only one node.
  - Some systems allow pre-splitting.
- Dynamic partition works for both key range and hash partitioned data.

Automatic or Manual Rebalancing

- Mostly manual → Why?
- Rebalancing is very expensive!
- Some suggestive interfaces exist.

Partitioning challenges

- How to partition and how to index?
- How to add or remove nodes?
- How to route the requests and execute queries?



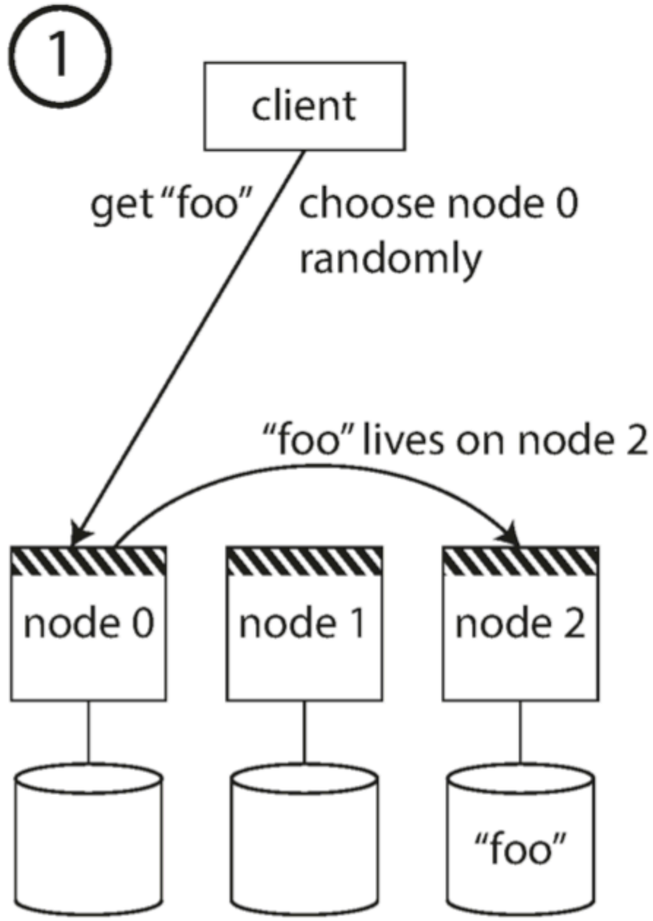
Shared-Nothing Parallelism

Two questions

- Which node to connect to?
- Where to maintain the knowledge of rebalanced results?

Routing paradigm #1

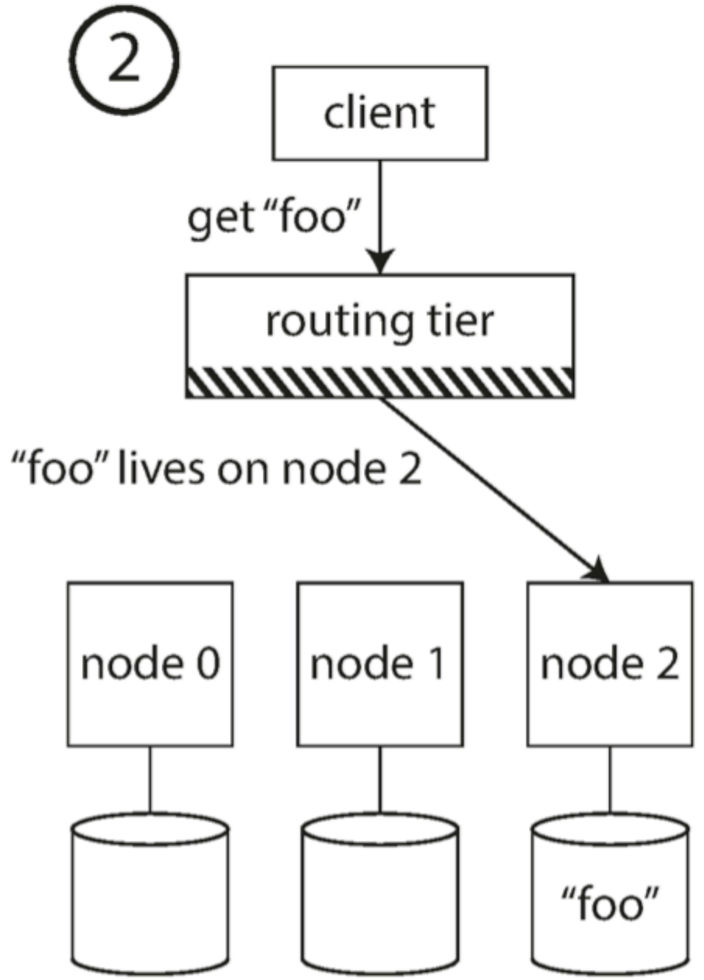
- Contact any node (e.g., a round-robin load balancer),
- If the node has the data copy, respond.
- If not, forward, receives the reply, and passes the reply along to the client.



▨ = the knowledge of which partition is assigned to which node

Routing paradigm #2

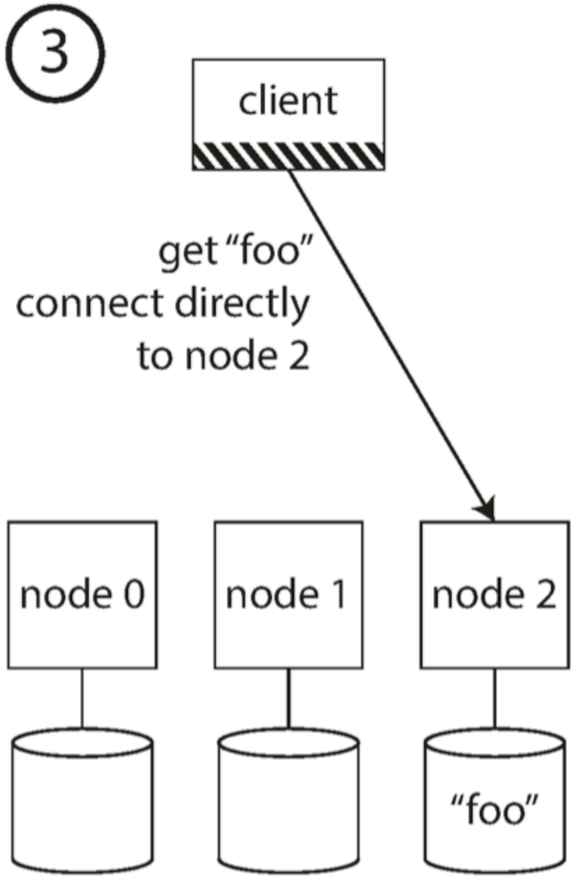
- Send all requests to a routing tier first.
- The routing tier forward all the requests.



▨ = the knowledge of which partition is assigned to which node

Routing paradigm #3

- The client is aware of the partitioning and the assignment of partitions to nodes.
- No intermediary.

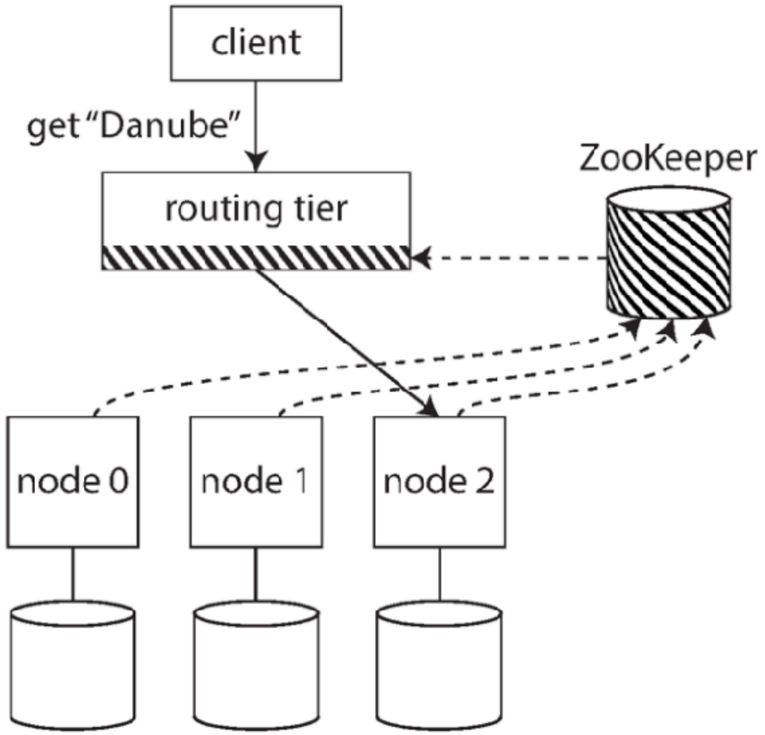


= the knowledge of which partition is assigned to which node

Two questions

- Which node to connect to?
- Where to maintain the knowledge of rebalanced results?

ZooKeeper

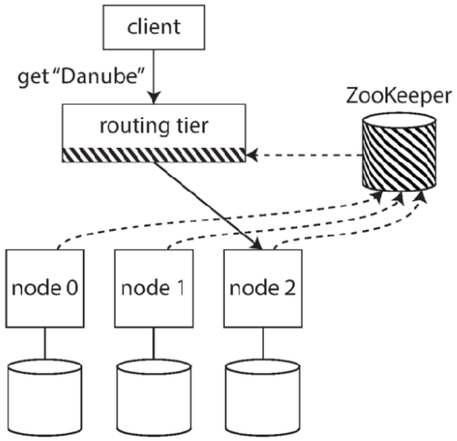


Key range	Partition	Node	IP address
A-ak — Bayes	partition 0	node 0	10.20.30.100
Bayeu — Ceanothus	partition 1	node 1	10.20.30.101
Ceara — Deluc	partition 2	node 2	10.20.30.102
Delusion — Frenssen	partition 3	node 0	10.20.30.100
Freon — Holderlin	partition 4	node 1	10.20.30.101
Holderness — Krasnoje	partition 5	node 2	10.20.30.102
Krasnokamsk — Menadra	partition 6	node 0	10.20.30.100
Menage — Ottawa	partition 7	node 1	10.20.30.101
Otter — Rethimnon	partition 8	node 2	10.20.30.102
Reti — Solovets	partition 9	node 0	10.20.30.100
Solovyov — Truck	partition 10	node 1	10.20.30.101
Trudeau — Zywiec	partition 11	node 2	10.20.30.102

= the knowledge of which partition is assigned to which node

ZooKeeper

- Each node registers itself in ZooKeeper.
- ZooKeeper maintains the mapping.
- Other actors (different in three paradigms) subscribe to ZooKeeper.
- Whenever the partition mapping changes, ZooKeeper notifies actors.



Key range	Partition	Node	IP address
A-ak — Bayes	partition 0	node 0	10.20.30.100
Bayeu — Ceanothus	partition 1	node 1	10.20.30.101
Ceara — Deluc	partition 2	node 2	10.20.30.102
Delusion — Frenssen	partition 3	node 0	10.20.30.100
Freon — Holderlin	partition 4	node 1	10.20.30.101
Holderness — Krasnoje	partition 5	node 2	10.20.30.102
Krasnokamsk — Menadra	partition 6	node 0	10.20.30.100
Menage — Ottawa	partition 7	node 1	10.20.30.101
Otter — Rethimnon	partition 8	node 2	10.20.30.102
Reti — Solovets	partition 9	node 0	10.20.30.100
Solovyov — Truck	partition 10	node 1	10.20.30.101
Trudeau — Zywiec	partition 11	node 2	10.20.30.102

= the knowledge of which partition is assigned to which node

Takeaway

- The benefits of Partitioning and Replication.
- The challenges of Partitioning and Replication.
- The tradeoffs of different strategies.
- Replication: single-leader, multiple-leader, leaderless
- Partition: Key range, hash, hybrid.
  - Partition rebalancing strategies: fixed, dynamic
  - Partition routing, ZooKeeper