RAMCloud: A Low-Latency Datacenter Storage System

Ankita Kejriwal

Stanford University

(Joint work with Diego Ongaro, Ryan Stutsman, Steve Rumble,

Mendel Rosenblum and John Ousterhout)

What if you had...

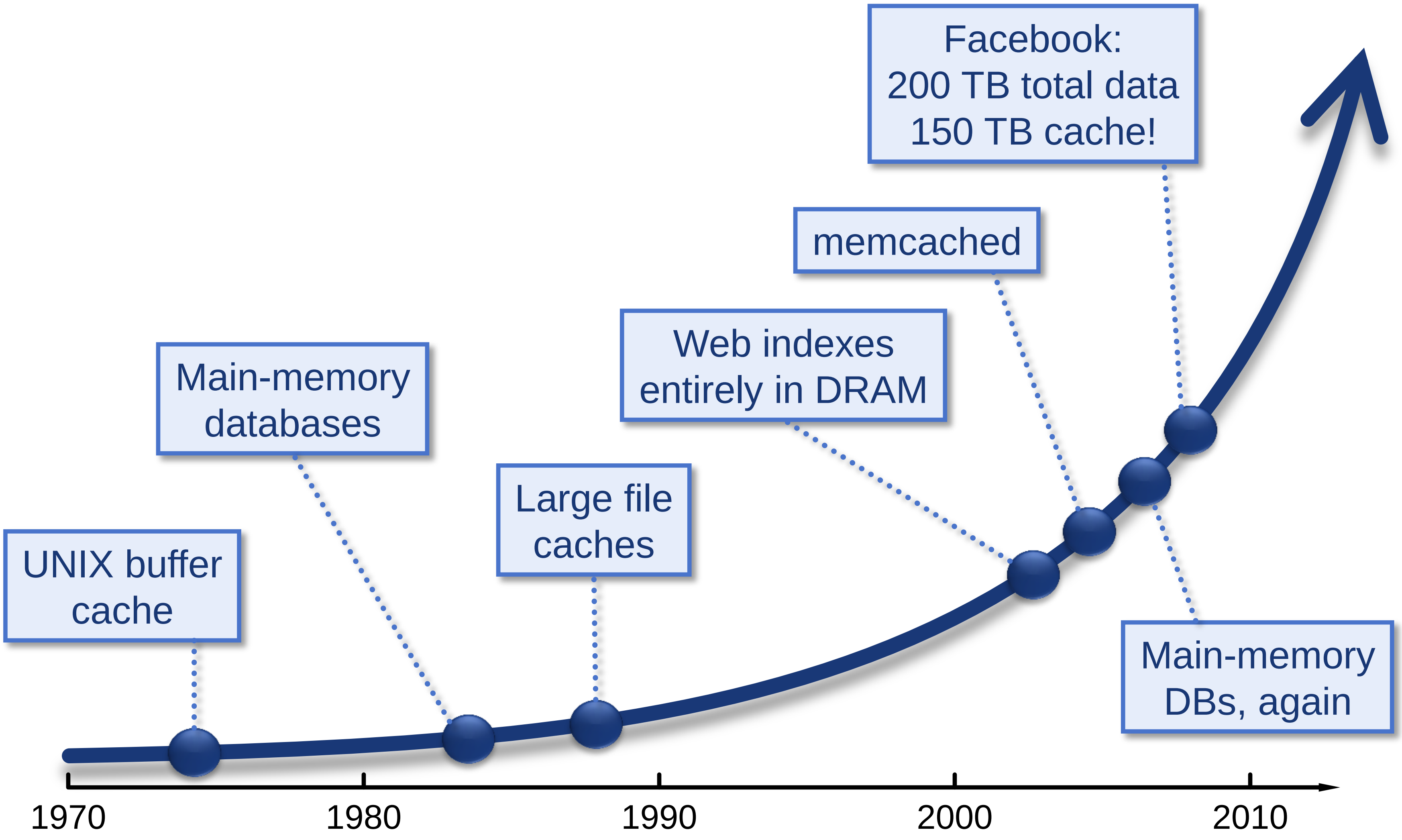
… a Storage System that provides:

- **Scale**
  - Data size: 10 PB
  - Accessible by 100,000 nodes (10 Million cores)
- **Uniform fast random access time to all data**
  - 100 B read: 2 µs RPC
  - 100 B write: 5 µs RPC
- **Durable and available**

RAMCloud

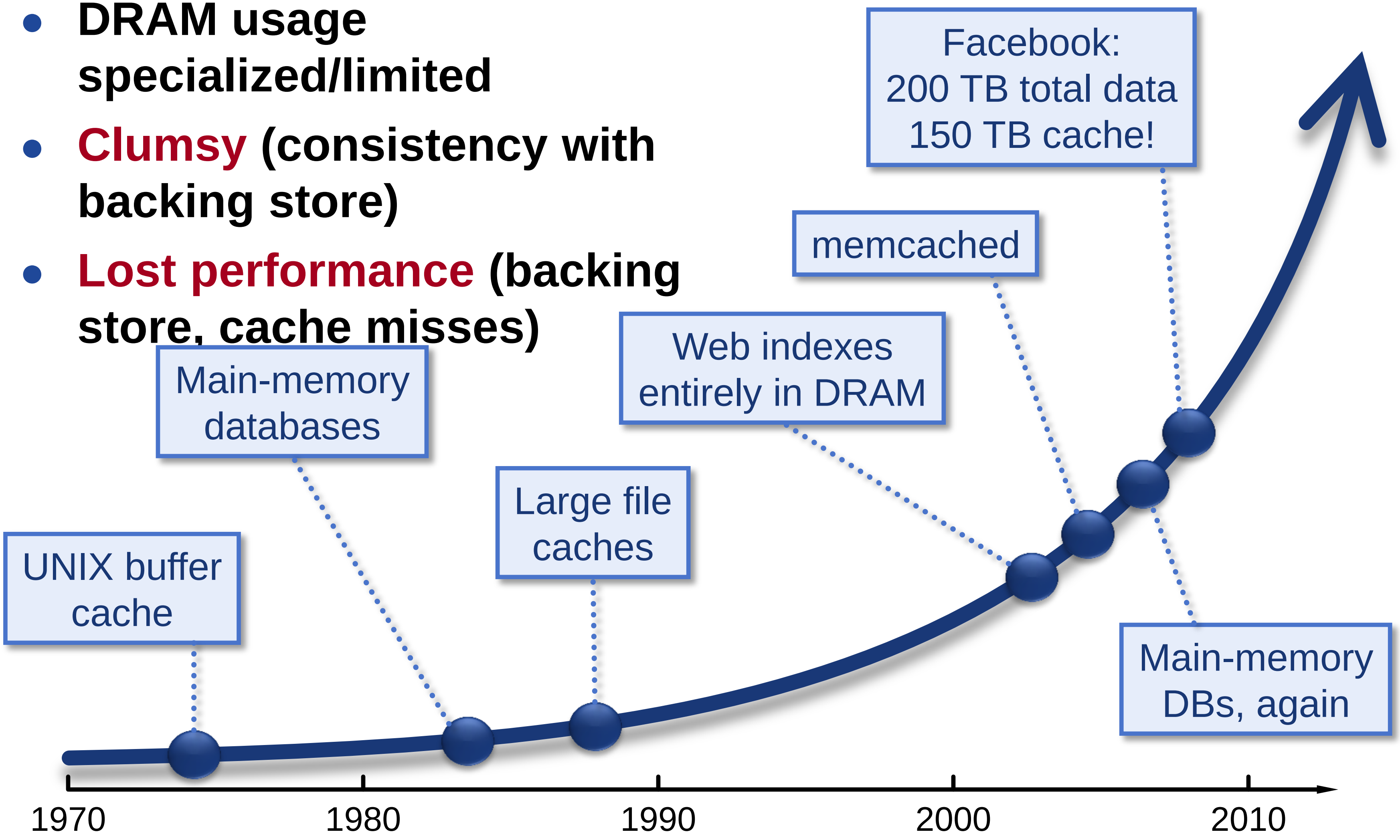
- **General-purpose storage system**
- **All data always in DRAM**
- **Scale: 1000 – 10000 servers, 1 PB data**
- **Performance goals:**
  - High throughput: 1M ops/sec/server
  - Low-latency access: 5-10µs RPC
- **Durable and available**
- **Potential impact: enable new class of applications**
  - Primary motivation: Web sphere
  - Maybe HPC?

DRAM in Storage Systems



DRAM in Storage Systems

- **DRAM usage specialized/limited**
- **Clumsy (consistency with backing store)**
- **Lost performance (backing store, cache misses)**



UNIX buffer cache

Main-memory databases

Large file caches

Web indexes entirely in DRAM

memcached

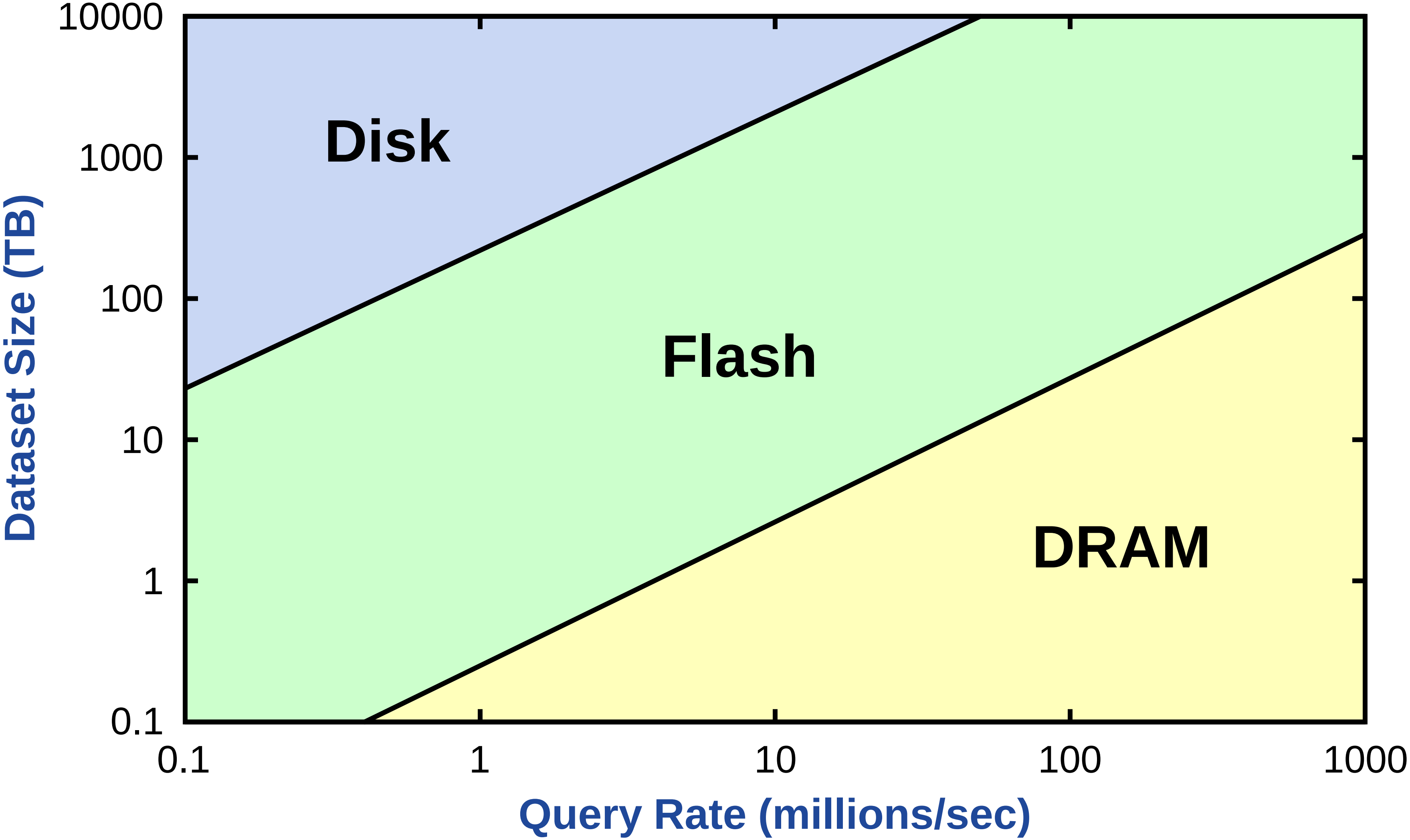
Facebook:
200 TB total data
150 TB cache!

Main-memory DBs, again

DRAM is cheaper!

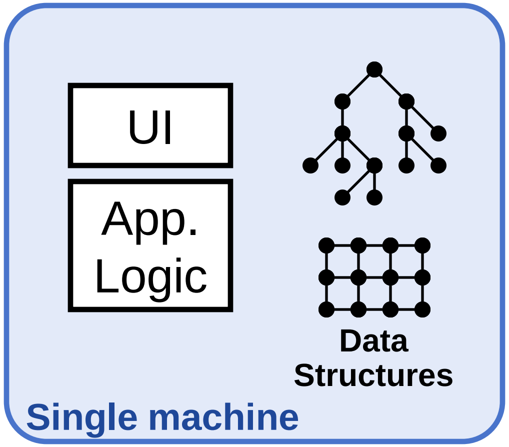
Lowest TCO

from "Andersen et al., "FAWN: A Fast Array of Wimpy Nodes", Proc. 22nd Symposium on Operating System Principles, 2009, pp. 1-14.



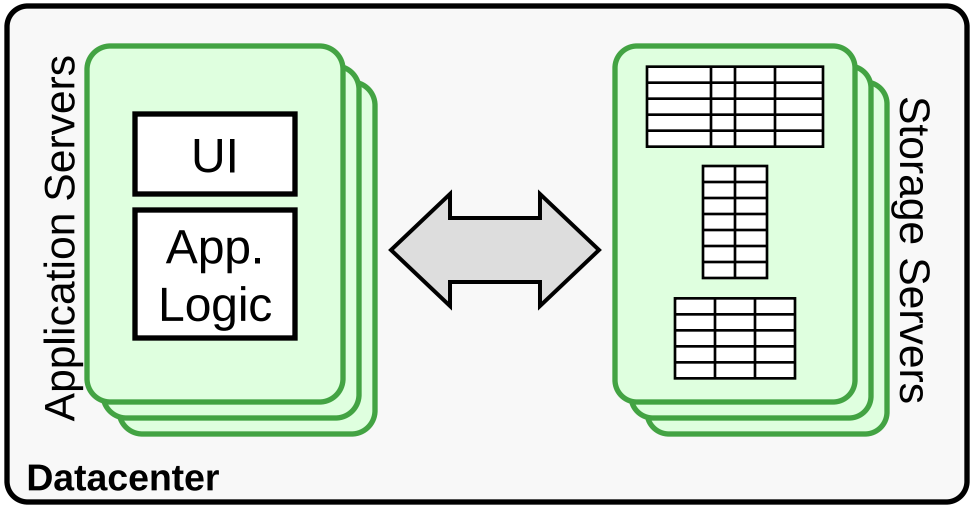
Why Does Latency Matter?

Traditional Application



<< 1µs latency

Web Application

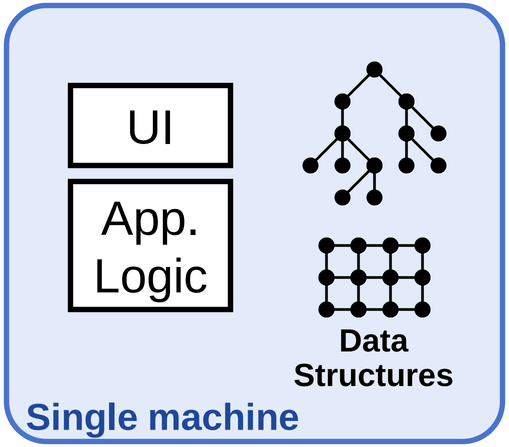


0.5-10ms latency

- **Large-scale apps struggle with high latency**
  - Random access data rate has not scaled!
  - Facebook: can only make 100-150 internal requests per page

RAMCloud Goal: Scale and Latency

Traditional Application



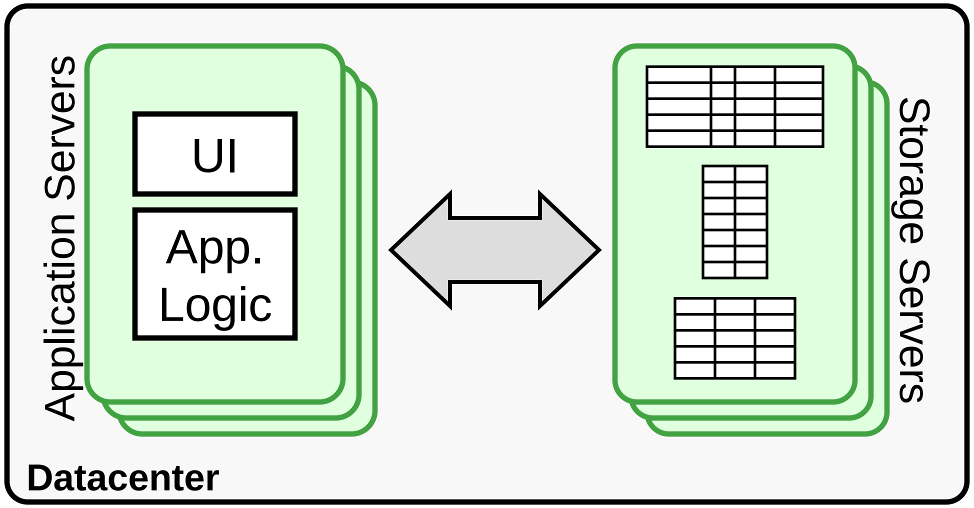
UI

App. Logic

Data Structures

Single machine

Web Application



Application Servers

UI

App. Logic

Storage Servers

Datacenter

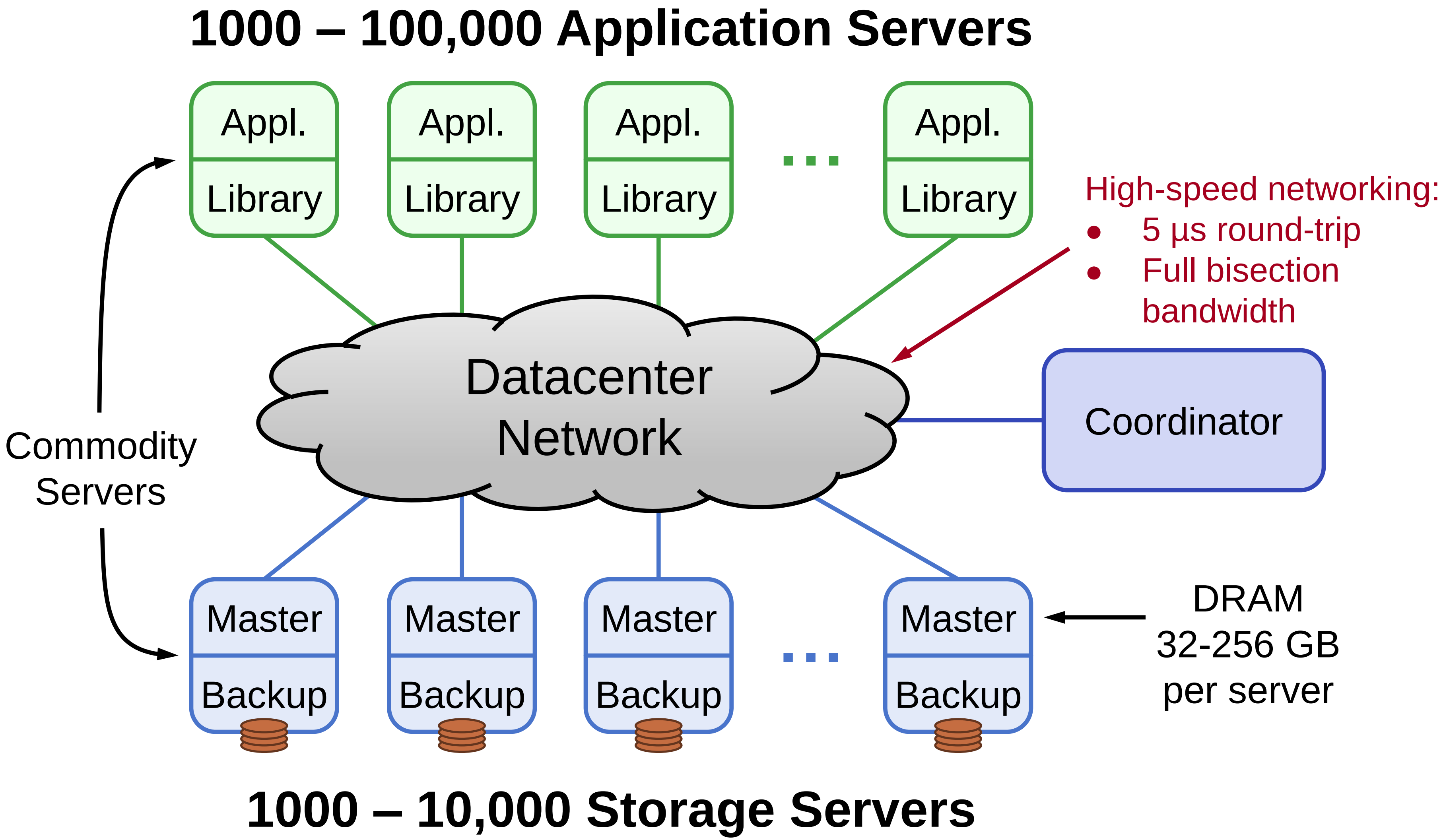
<< 1µs latency

~~0.5-10ms~~ latency

5-10µs

- **Enable new class of applications**

RAMCloud Architecture



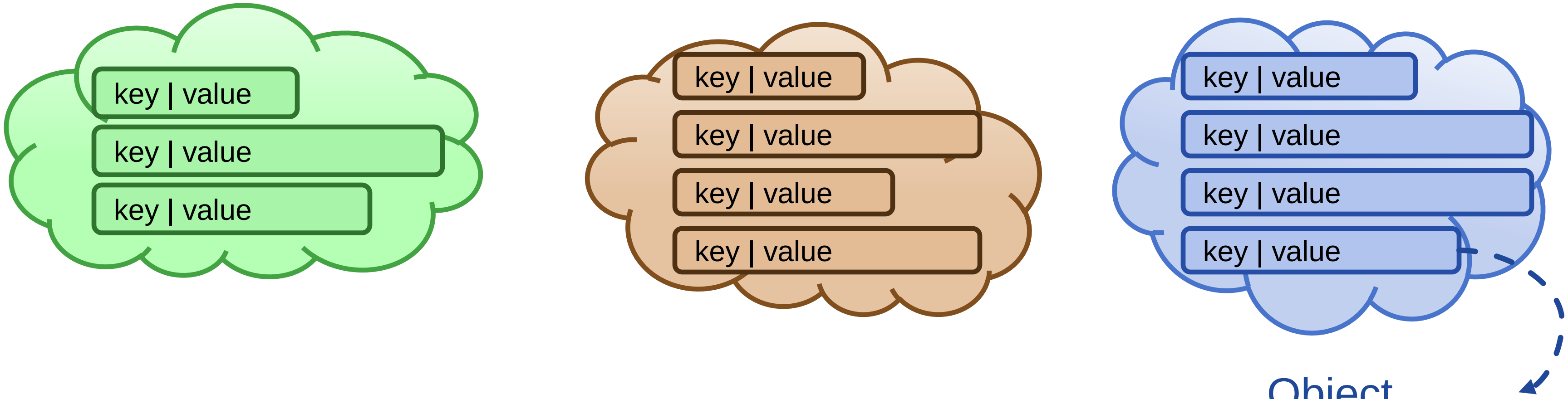
High-speed networking:

- 5 µs round-trip
- Full bisection bandwidth

DRAM
32-256 GB
per server

Data Model: Key-Value Store

Tables



Object

Key (≤ 64KB)

Version (64b)

Blob (≤ 1MB)

```
read(tableId, key)
    => blob, version

write(tableId, key, blob)
    => version

cwrite(tableId, key, blob, version)
    => version

delete(tableId, key)

enumerate(tableId)
```

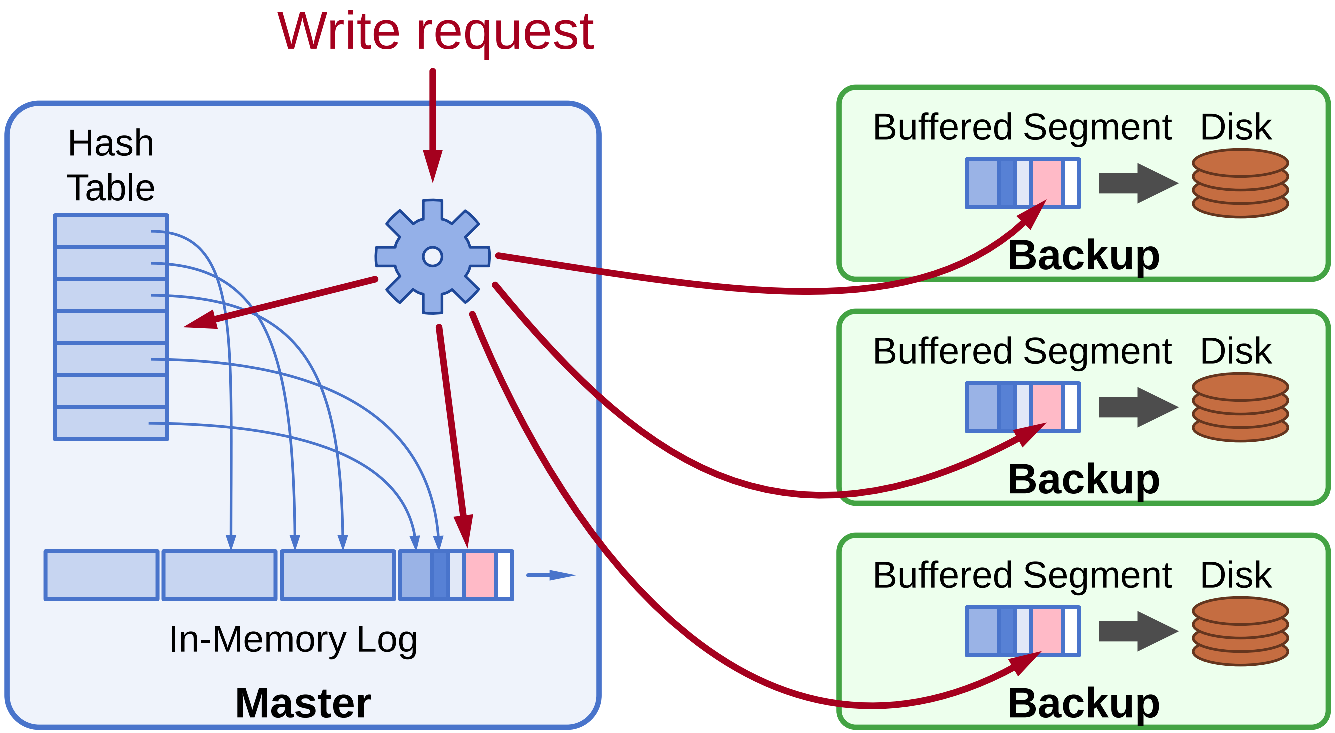
Richer model in the future:

- Indexes?
- Transactions?
- Graphs?

Durability and Availability

- **Goals:**
  - No impact on performance
  - Minimum cost, energy
- **Keep replicas in DRAM of other servers?**
  - 3x system cost, energy
  - Still have to handle power failures
- **RAMCloud approach:**
  - 1 copy in DRAM
  - Backup copies on disk/flash: durability ~ free!
- **Issues to resolve:**
  - Synchronous disk I/O’s during writes??
  - Data unavailable after crashes??

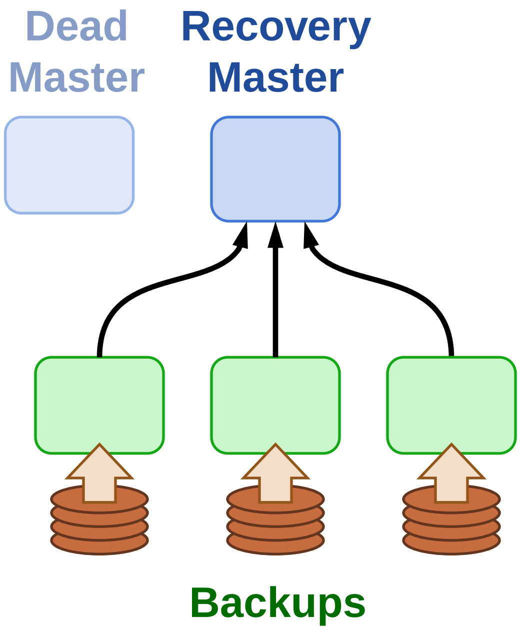
Buffered Logging



- No disk I/O during write requests
- Log-structured: backup disks and master's memory
- Log cleaning

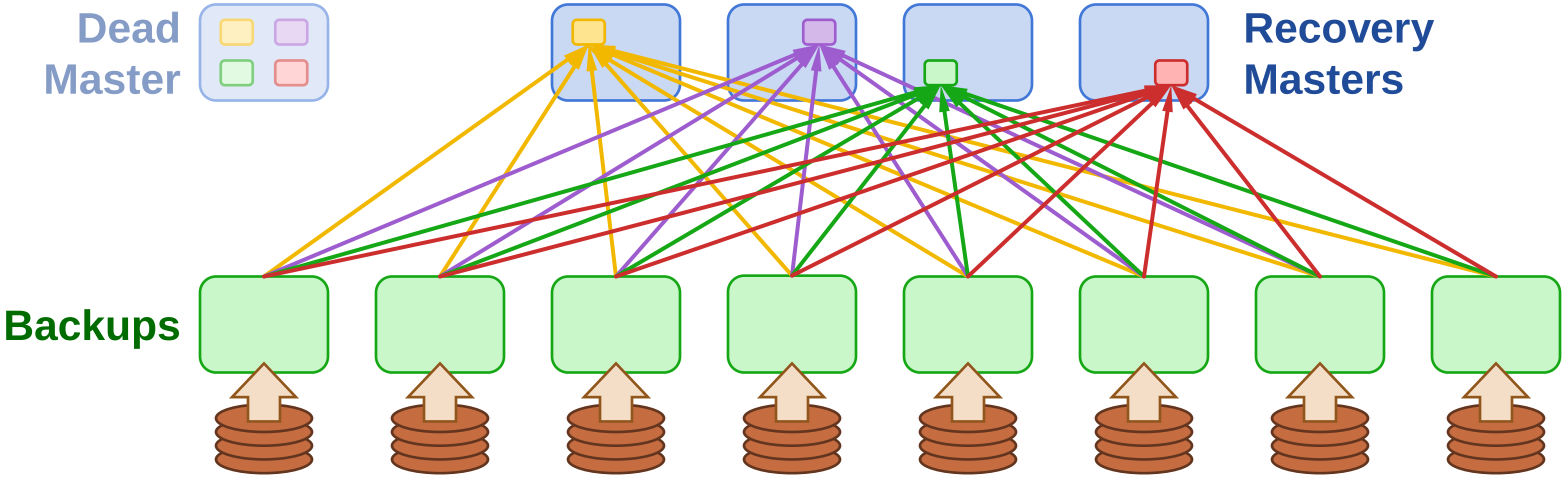
Crash Recovery

- **Server crashes:**
  - Must replay log to reconstruct data
- **Crash recovery:**
  - Choose recovery master
  - Backup reads log info from disk
  - Transfers logs to recovery master
  - Recovery master replays log
- **Meanwhile, data is unavailable**
- **RAMCloud approach: fast crash recovery**
  - 1-2 seconds for 100 GB of data
  - Use system scale to get around bottlenecks



Fast Crash Recovery

- **Scatter backup data across backups**
- **Divide each master's data into partitions**
  - Recover each partition on a separate recovery master
  - Each backup divides its log data among recovery masters



RAMCloud Project Status

- **Goal: build production-quality implementation**
- **Nearing 1.0-level release**
- **Current test cluster:**
  - 80 servers, 2 TB data
  - High speed Infiniband networking
  - Performance:
    - 100 B read: 5.3 µs RPC
    - 100 B write: 15 µs RPC
- **Interested in finding applications for RAMCloud**

Is RAMCloud right for HPC apps?

Properties of RAMCloud relevant to application developers:

- **Durability and availability**
- **Key-value store**
- **Commodity hardware**
- **Read / write access latency**
- **Random access to small objects**

Conclusion

- **General-purpose storage system**
- **All data always in DRAM**
- **Designed for:**
  - Scale: 1000 – 10000 servers, 1 PB data
  - Performance: 5-10µs RPC
- **Durable and available**

Questions

- **Is RAMCloud appropriate for HPC Applications?**
  - Durability and availability
  - Key-value store
  - Commodity hardware
  - Read / write access latency
  - Random access to small objects
- **One thing that we could change to make RAMCloud interesting to you!**

Thank you!