RAMCloud: Scalable Datacenter Storage Entirely in DRAM

John Ousterhout

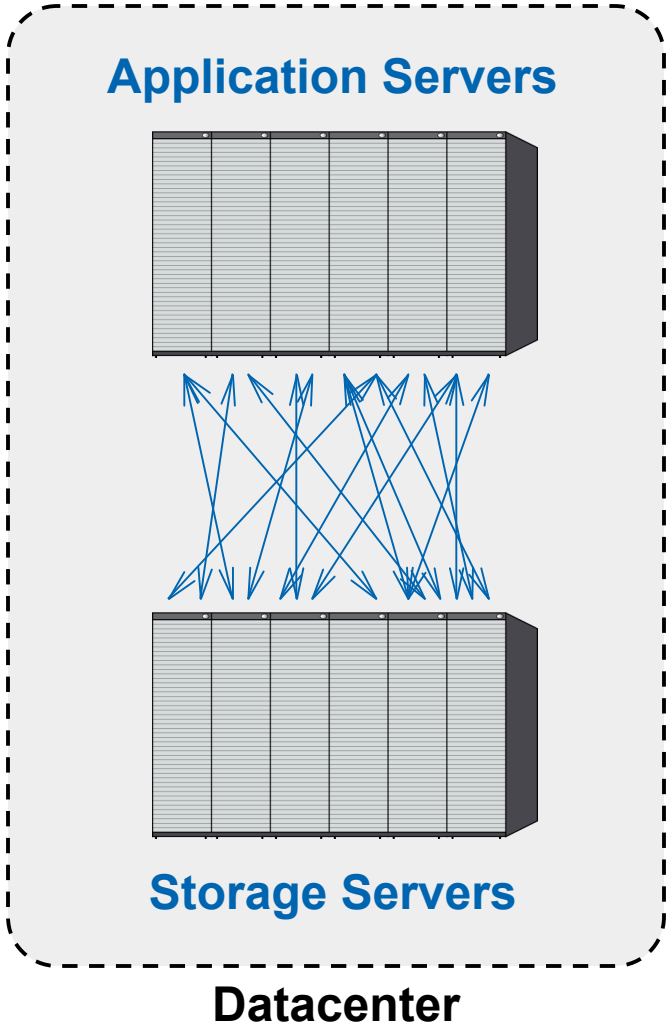
Stanford University

Introduction

- New research project at Stanford
- Create large-scale storage systems entirely in DRAM
- Interesting combination: scale, low latency
- The future of datacenter storage
- Low latency disruptive to database community

RAMCloud Overview

- Storage for datacenters
- 10-10000 commodity servers
- ~64 GB DRAM/server
- All data always in RAM
- Durable and available
- High throughput: 1M ops/sec/server
- Low-latency access: 5-10µs RPC



Example Configurations

	Today	5-10 years
# servers	1000	1000
GB/server	64GB	1024GB
Total capacity	64TB	1PB
Total server cost	$4M	$4M
$/GB	$60	$4

RAMCloud Motivation

- **Relational databases don't scale**
- **Every large-scale Web application has problems:**
  - Facebook: 4000 MySQL servers + 2000 memcached servers
- **New forms of storage starting to appear:**
  - Bigtable
  - Dynamo
  - PNUTS
  - H-store
  - memcached

RAMCloud Motivation, cont'd

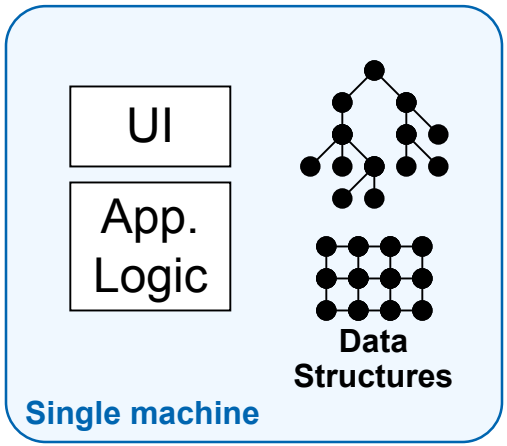
Disk access rate not keeping up with capacity:

	Mid-1980's	2009	Change
Disk capacity	30 MB	500 GB	16667x
Max. transfer rate	2 MB/s	100 MB/s	50x
Latency (seek & rotate)	20 ms	10 ms	2x
Capacity/bandwidth (large blocks)	15 s	5000 s	333x
Capacity/bandwidth (1KB blocks)	600 s	58 days	8333x
Jim Gray's rule	5 min	30 hrs	360x

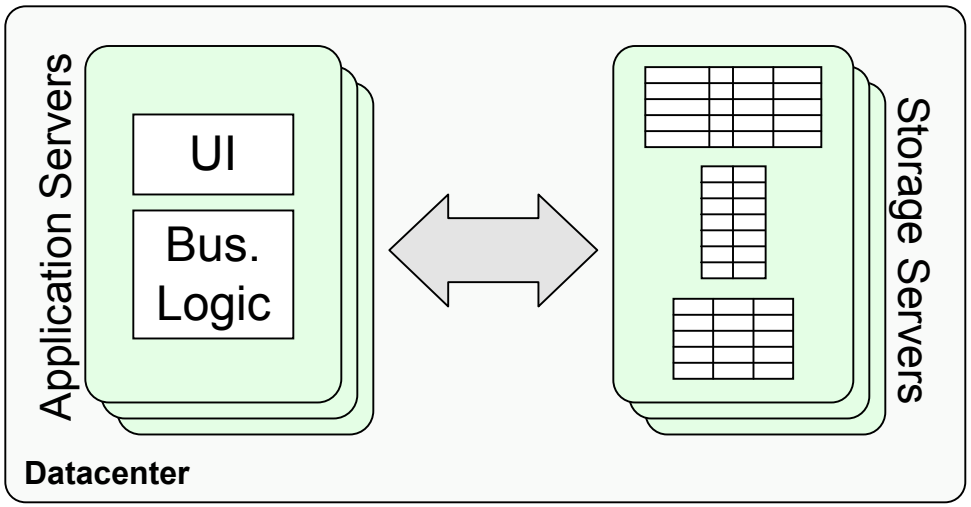
- **Disks must become more archival**
- **More information must move to memory**

Impact of Latency

Traditional Application



Web Application



<< 1µs latency

0.5-10ms latency

- **Large-scale apps struggle with high latency**
- **RAMCloud goal: low latency and large scale**
- **Enable a new breed of information-intensive applications**

Research Issues

- Achieving 5-10 µs RPC
- Durability at low latency
- Data model
- Concurrency/consistency model
- Data distribution, scaling
- Automated management
- Multi-tenancy
- Node architecture

Low Latency: SQL is Dead?

- **Relational query model tied to high latency:**
  - Describe what you need up fron
  - DBMS optimizes retrieval
- **With sufficiently low latency:**
  - Don't need optimization; make individual requests as needed
  - Can't afford query processing overhead
  - The relational query model will disappear
- **Question: what systems offer very low latency and use relational model?**

Low Latency: Stronger Consistency?

- **Cost of consistency rises with transaction overlap:**

  O ~ R*D

  O = # overlapping transactions
  R = arrival rate of new transactions
  D = duration of each transaction

- **R increases with system scale**
  - Eventually, scale makes consistency unaffordable
- **But, D decreases with lower latency**
  - Stronger consistency affordable at larger scale
  - Is this phenomenon strong enough to matter?

Low Latency: One Size Fits All Again?

- **"One-size-fits-all is dead" - Mike Stonebraker**
- **Specialized databases proliferating:**
  - 50x performance improvements in specialized domains
  - Optimize disk layout to eliminate seeks
- **With low latency:**
  - Layout doesn't matter
  - General-purpose is fast
  - One-size-fits-all rides again?

Conclusions

- **All online data is moving to RAM**
- **RAMClouds = the future of datacenter storage**
- **Low latency will change everything:**
  - New applications
  - Stronger consistency at scale
  - One-size-fits-all again
  - SQL is dead
- **1000-10000 clients accessing 100TB - 1PB @ 5-10µs latency**

Questions/Comments?

For more on RAMCloud motivation & research issues:

- "The Case for RAMClouds: Scalable High-Performance Storage Entirely in DRAM"
- To appear in *Operating Systems Review*
- http://www.stanford.edu/~ouster/cgi-bin/papers/ramcloud.pdf
- Or, google "RAMCloud"

Backup Slides

Why Not a Caching Approach?

- **Lost performance:**
  - 1% misses → 10x performance degradation
- **Won't save much money:**
  - Already have to keep information in memory
  - Example: Facebook caches ~75% of data size
- **Changes disk management issues:**
  - Optimize for reads, vs. writes & recovery

Why not Flash Memory?

- **Many candidate technologies besides DRAM**
  - Flash (NAND, NOR)
  - PC RAM
  - …
- **DRAM enables lowest latency:**
  - 5-10x faster than flash
- **Most RAMCloud techniques will apply to other technologies**
- **Ultimately, choose storage technology based on cost, performance, energy, not volatility**

Is RAMCloud Capacity Sufficient?

- **Facebook: 200 TB of (non-image) data today**
- **Amazon:**

Revenues/year:	$16B
Orders/year:	400M? ($40/order?)
Bytes/order:	1000-10000?
Order data/year:	0.4-4.0 TB?
RAMCloud cost:	$24K-240K?

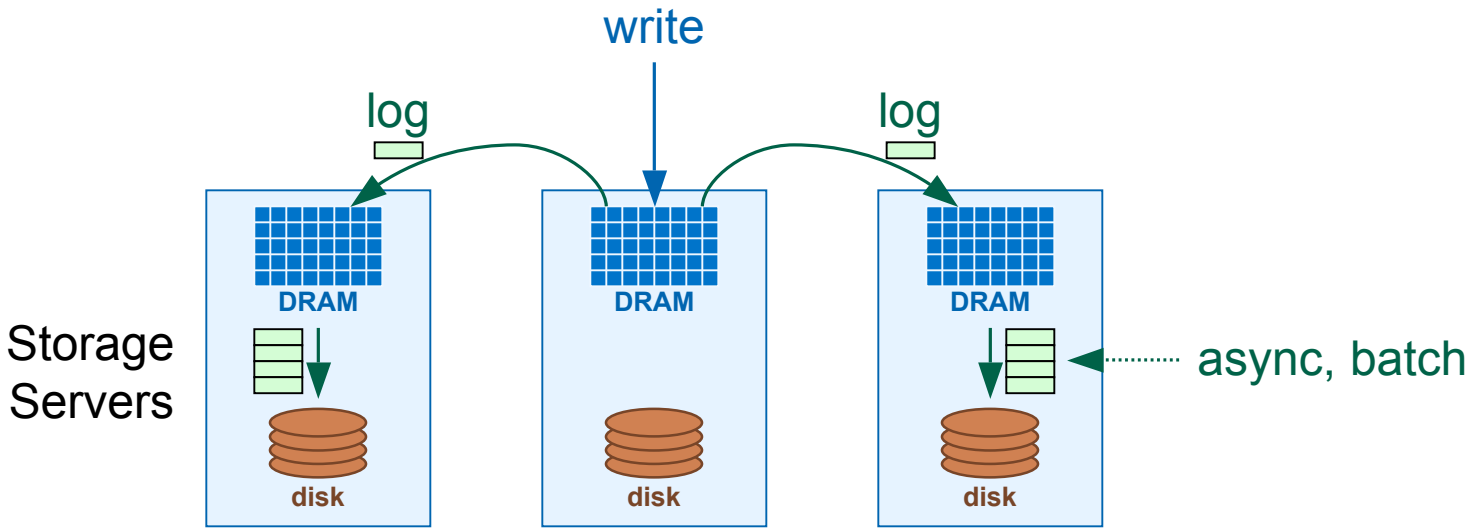
- **United Airlines:**

Total flights/day:	4000? (30,000 for all airlines in U.S.)
Passenger flights/year:	200M?
Bytes/passenger-flight:	1000-10000?
Order data/year:	0.2-2.0 TB?
RAMCloud cost:	$13K-130K?

- **Ready today for all online data; media soon**

Data Durability/Availability

- **Data must be durable when write RPC returns**
- **Unattractive possibilities:**
  - Synchronous disk write (100-1000x too slow)
  - Replicate in other memories (too expensive)
- **One possibility: log to RAM, then disk**



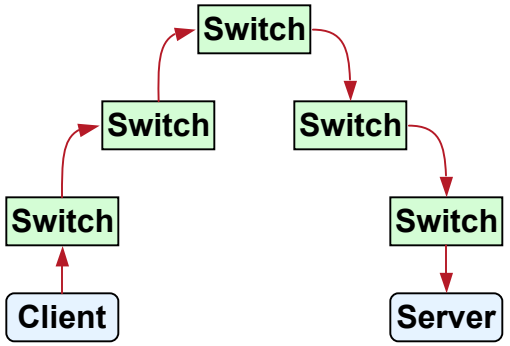
Durability/Availability, cont'd

- **Buffered logging supports ~50K writes/sec./server (vs. 1M reads)**
- **Need fast recovery after crashes:**
  - Read 64 GB from disk? 10 minutes
  - Shard backup data across 100's of servers
  - Reduce recovery time to 1-2 seconds
- **Other issues:**
  - Power failures
  - Cross-datacenter replication

Low-Latency RPCs

Achieving 5-10µs will impact every layer of the system:

- **Must reduce network latency:**
  - Typical today: 10-30 µs/switch, 5 switches each way
  - Arista: 0.9 µs/switch: 9 µs roundtrip
  - Need cut-through routing, congestion mgmt



- **Tailor OS on server side:**
  - Dedicated cores
  - No interrupts?
  - No virtual memory?

Low-Latency RPCs, cont'd

- **Client side: need efficient path through VM**
  - User-level access to network interface?
- **Network protocol stack**
  - TCP too slow (especially with packet loss)
  - Must avoid copies
- **Preliminary experiments:**
  - 10-15 µs roundtrip
  - Direct connection: no switches

Interesting Facets

- **Use each system property to improve the others**
- **High server throughput:**
  - No replication for performance, only durability?
  - Simplifies consistency issues
- **Low-latency RPC:**
  - Cheap to reflect writes to backup servers
  - Stronger consistency?
- **1000's of servers:**
  - Sharded backups for fast recovery

New Conference!

USENIX Conference on Web Application Development:

- All topics related to developing and deploying Web applications
- First conference: June 20-25, 2010, Boston
- Paper submission deadline: January 11, 2010
- http://www.usenix.org/events/webapps10/