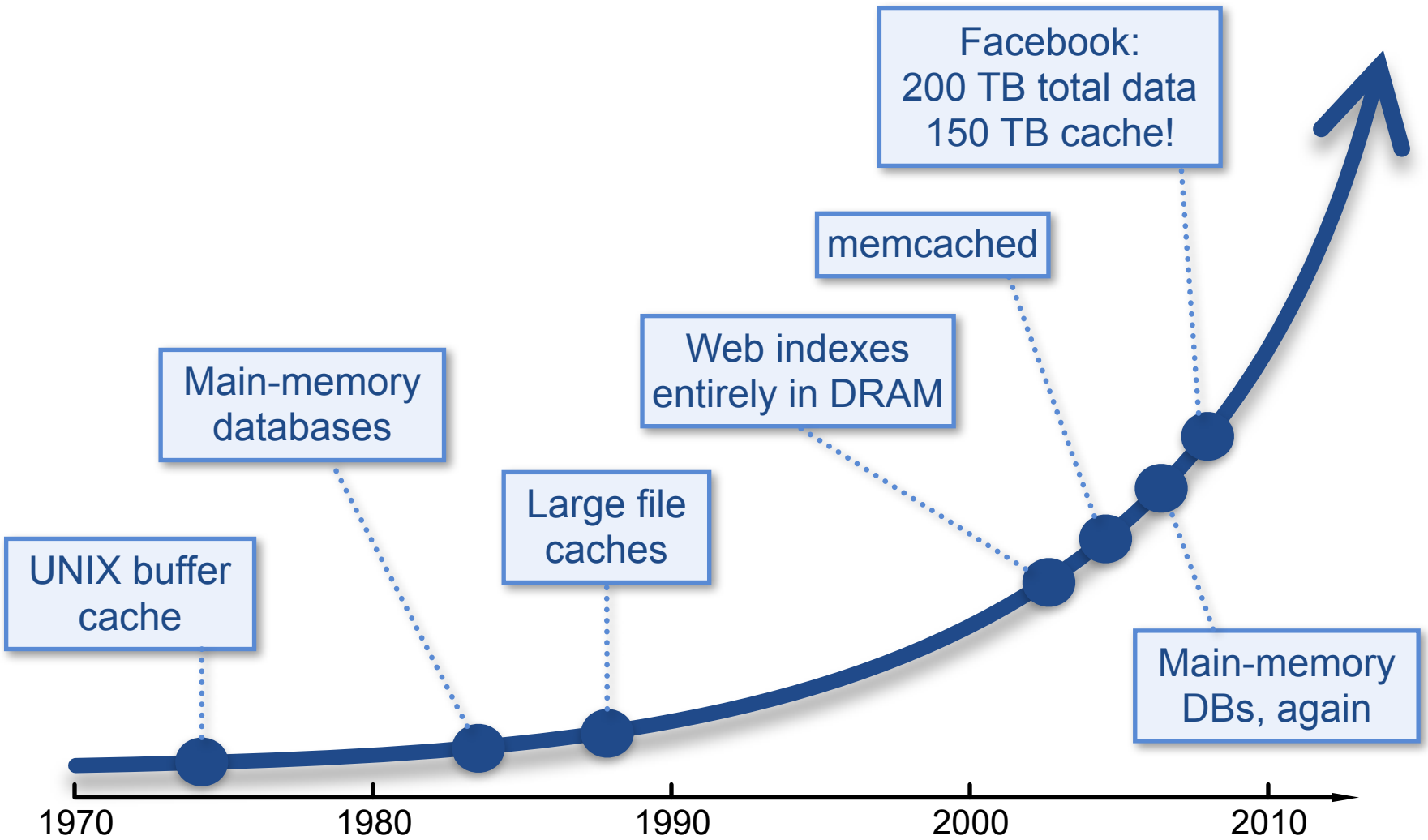
RAMCloud: The Future of Datacenter Storage?

John Ousterhout
Stanford University

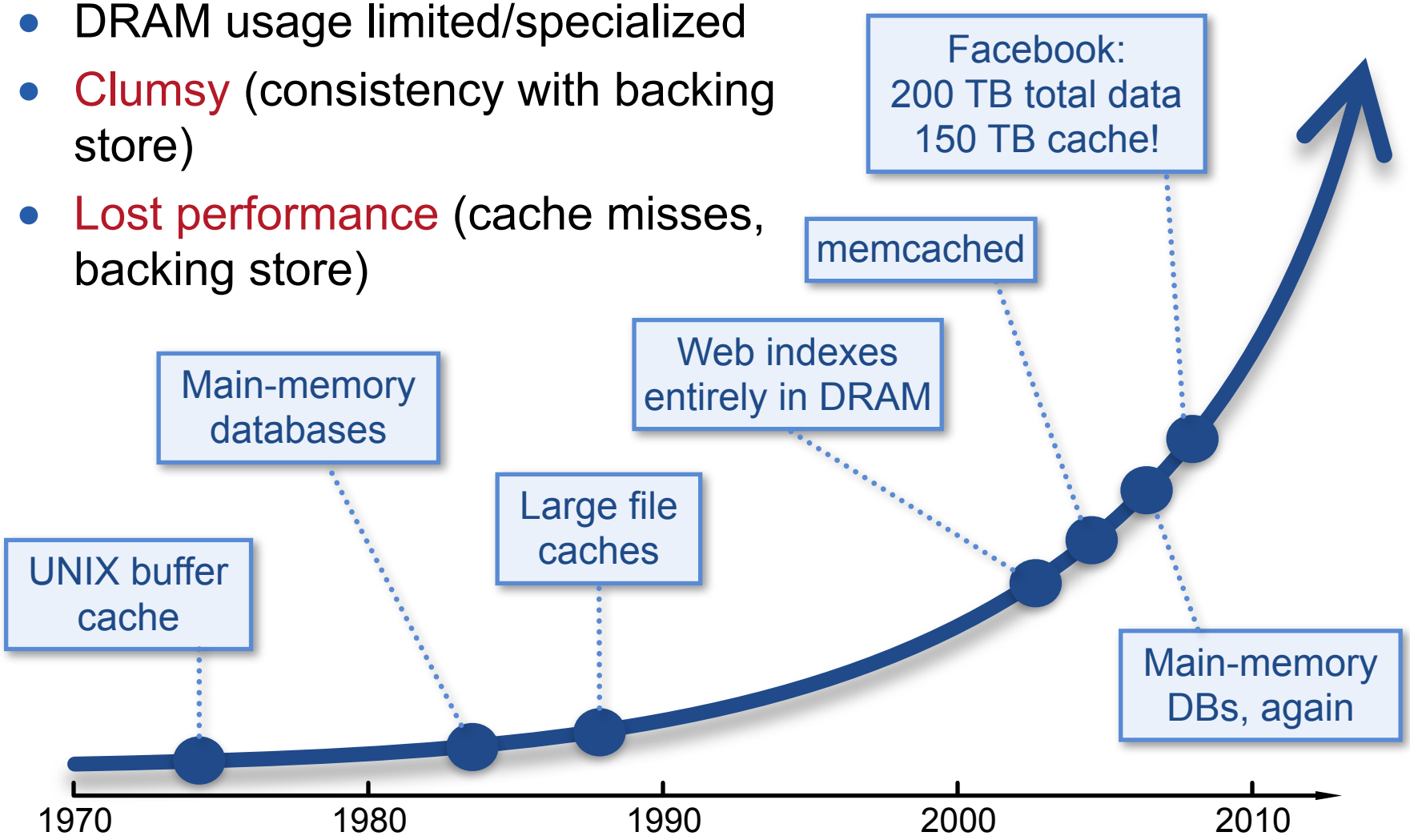
(with Nandu Jayakumar, Ankita Kejriwal, Diego Ongaro,
Mendel Rosenblum, Stephen Rumble, and Ryan Stutsman)

DRAM in Storage Systems



DRAM in Storage Systems

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



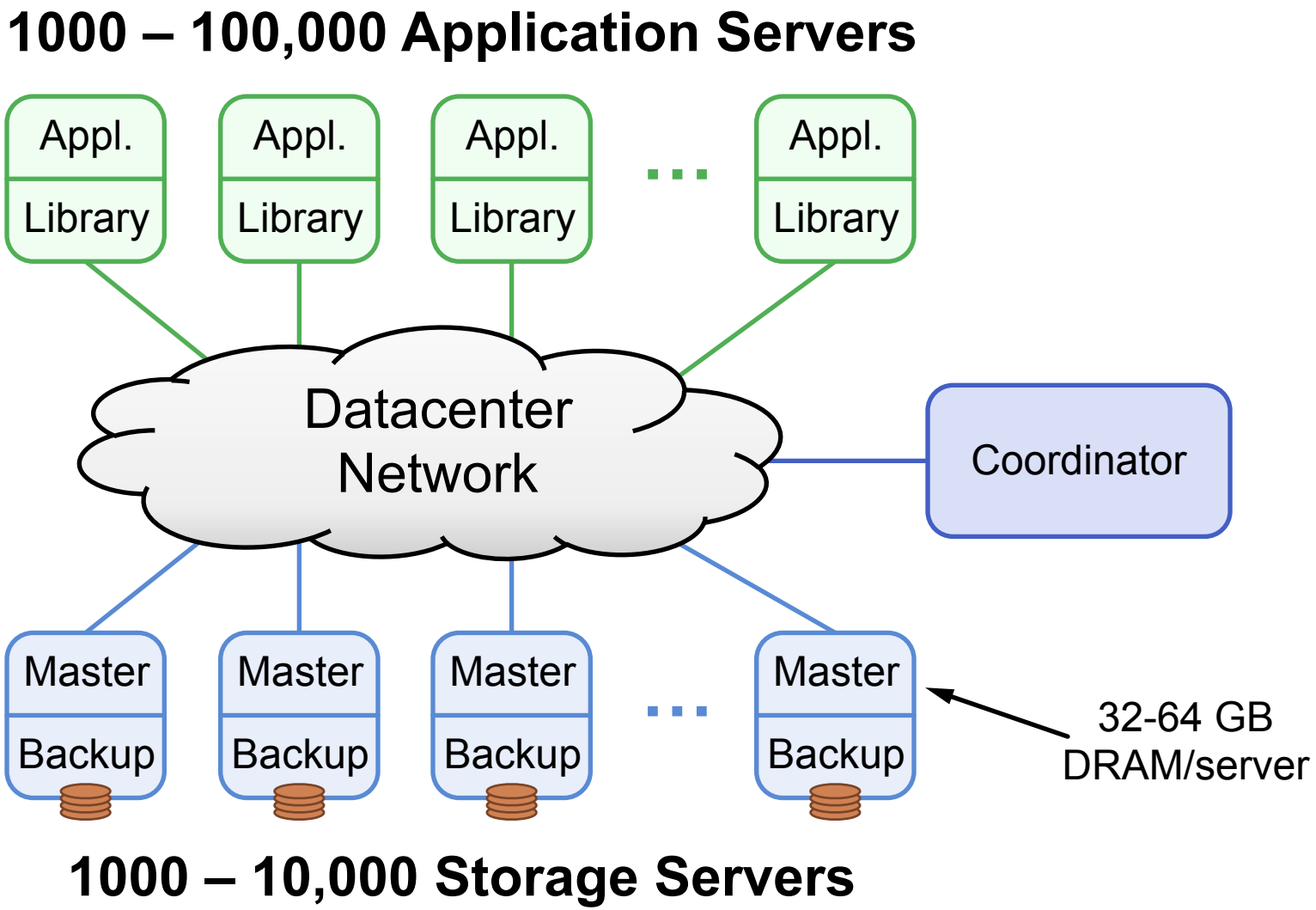
RAMCloud

Harness full performance potential of large-scale DRAM storage:

- General-purpose storage system
- All data always in DRAM (no cache misses)
- Durable and available (no backing store)
- Scale: 1000+ servers, 100+ TB
- Low latency: 5-10µs remote access

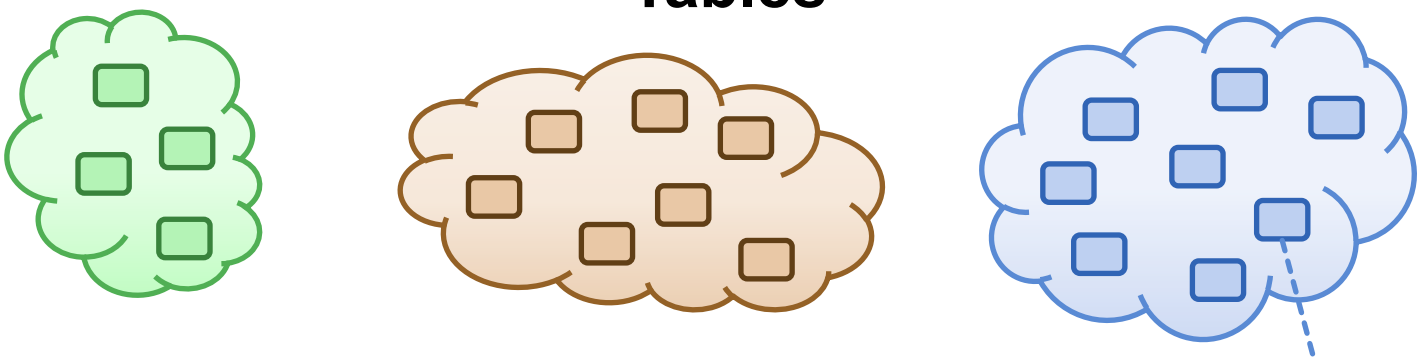
Potential impact: enable new class of applications

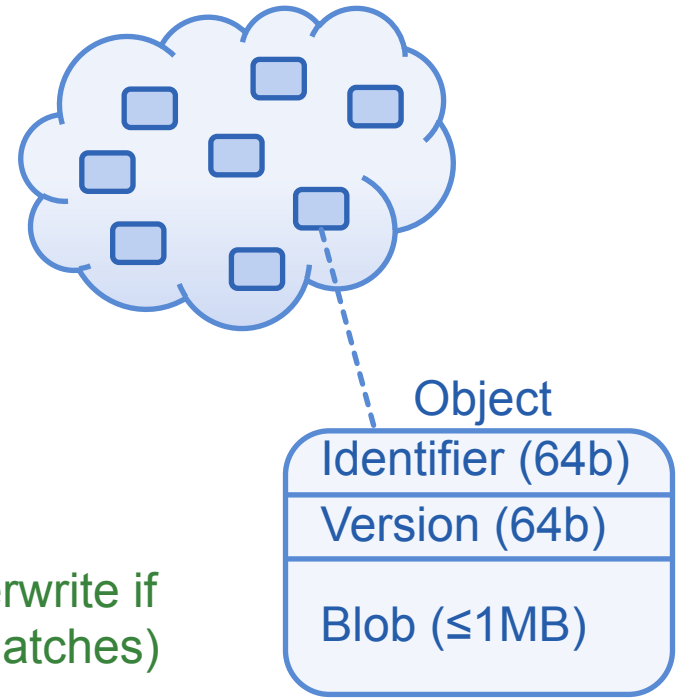
RAMCloud Architecture



Data Model

Tables






```
create(tableId, blob)
    => objectId, version

read(tableId, objectId)
    => blob, version

write(tableId, objectId, blob)
    => version

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

delete(tableId, objectId)
```

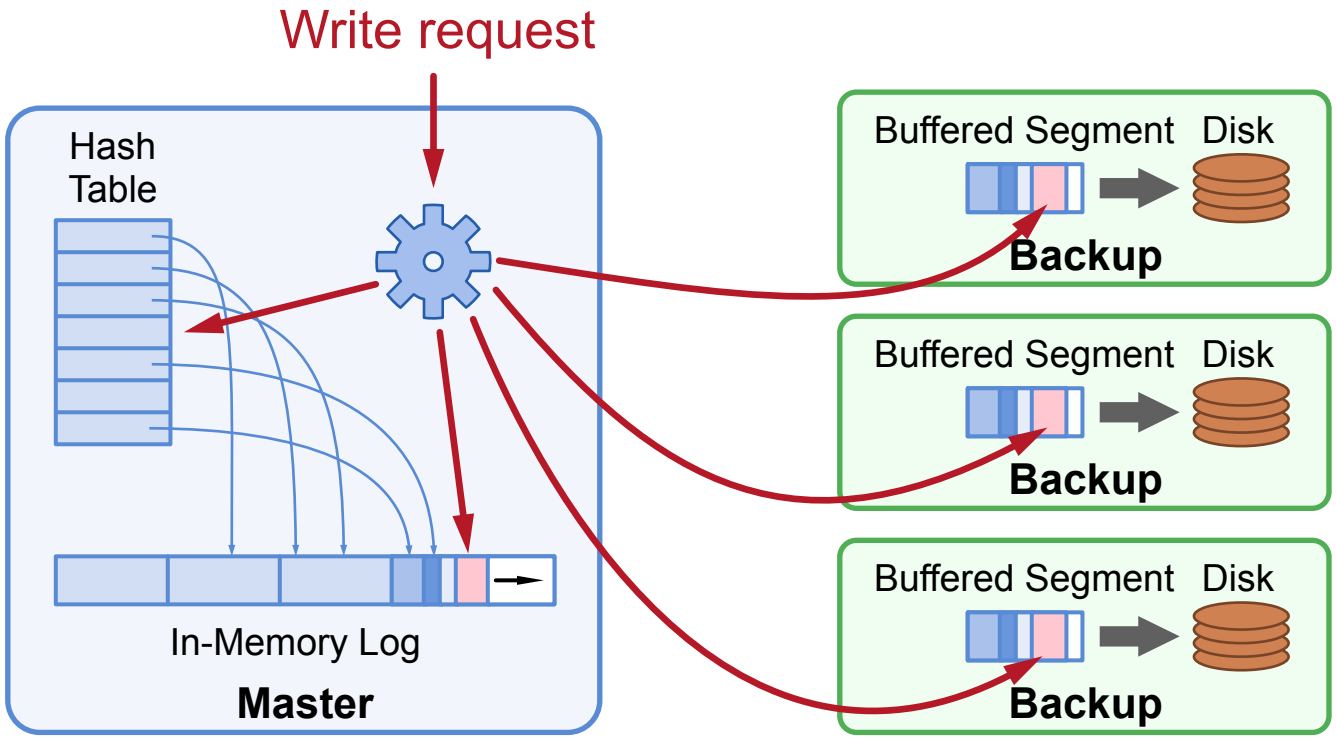
(Only overwrite if version matches)

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
  - Replicas unnecessary for performance
- 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



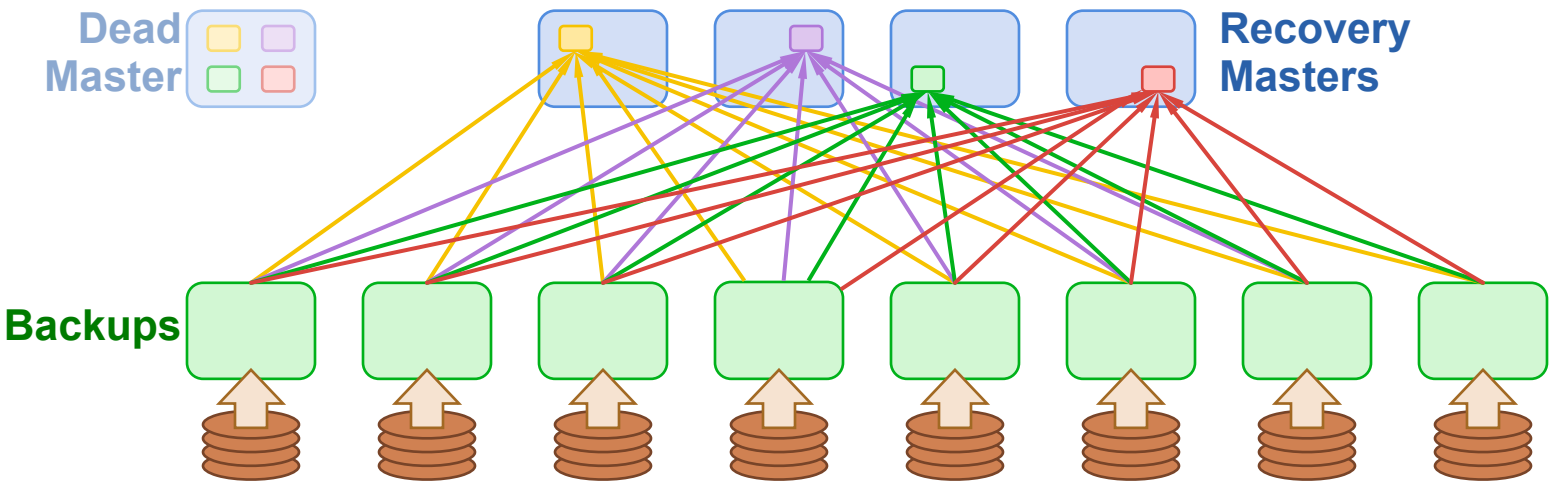
- No disk I/O during write requests
- Master's memory also log-structured
- Log cleaning ~ generational garbage collection

Crash Recovery

- Power failures: backups must guarantee durability of buffered data:
  - DIMMs with built-in flash backup
  - Per-server battery backups
  - Caches on enterprise disk controllers
- Server crashes:
  - Must replay log to reconstruct data
  - Meanwhile, data is unavailable
  - Solution: fast crash recovery (1-2 seconds)
  - If fast enough, failures will not be noticed
- Key to fast recovery: use system scale

Fast Recovery

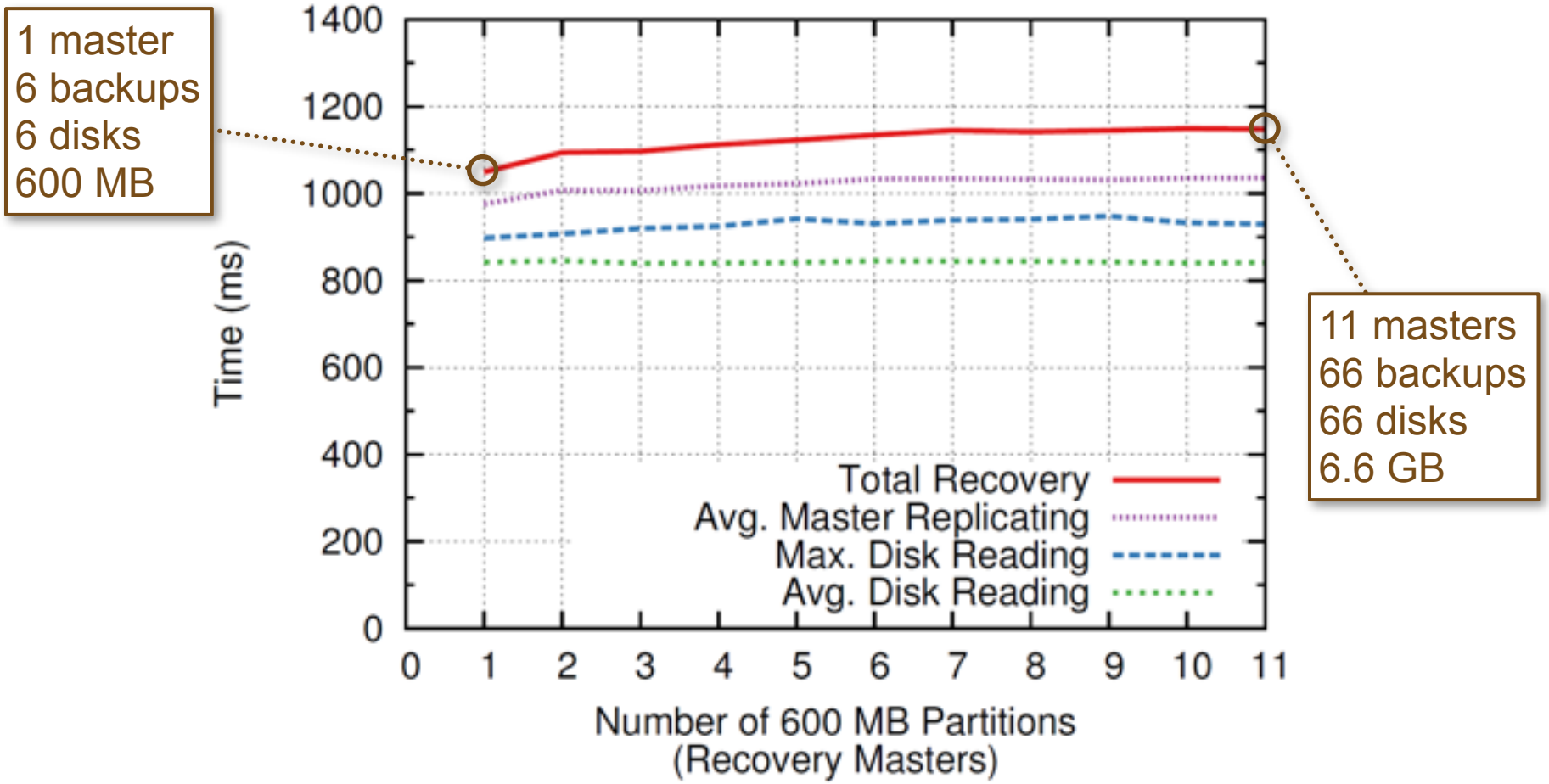
- Scatter backup data for each master
  - Use ~1000 disks during recovery
- Divide each master's data into partitions
  - Recover each partition on a separate recovery master
  - Use many NICs, many CPUs



Project Status

- Goal: build production-quality implementation
- Started coding Spring 2010
- Major pieces coming together:
  - RPC subsystem
    - Supports many different transport layers
    - Using Mellanox Infiniband for high performance
  - Basic data model
  - Simple cluster coordinator
  - Fast recovery
- Performance (40-node cluster):
  - Read small object: 5µs
  - Throughput: ~ 1M small reads/second/server

Recovery Scalability



Conclusion

- Results so far:
  - Achieved low latency (at small scale)
  - 1 second crash recovery for memory sizes < 10GB
  - Durable and available DRAM storage for the cost of volatile cache
- Lots more work still to do:
  - Not yet at large scale (but scalability encouraging)
  - Many other research issues (fast RPC, data model, …)
- Potential long-term impact:
  - Harness full performance potential of DRAM-based storage
  - Enable new applications: intensive manipulation of large-scale data
  - The future of datacenter storage?

Example Configurations

	Today	5-10 years
# servers	2000	4000
GB/server	24GB	256GB
Total capacity	48TB	1PB
Total server cost	$3.1M	$6M
$/GB	$65	$6

For $100-200K today:

- One year of Amazon customer orders
- One year of United flight reservations

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
- Availability gaps after crashes:
  - System performance intolerable until cache refills
  - Facebook example: 2.5 hours to refill caches!

RAMCloud Motivation: Technology

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**

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