Oasis+ Distributed Storage System

UNIVERSITY OF CALIFORNIA

RIVERSIDE

David Watson, Yan Luo, Brett Fleisch

Oasis+ Rationale

- High availability memory-based system can be used to store client specific information on a server backbone
- Developing reliable (highly available) services is currently *very* difficult
  - current DB-based technologies are very expensive, and do not fit “object model” well;
  - cluster systems do not provide good high performance solutions yet
- DSM has been studied for several years and is very efficient but not deployed
- May be thought of as a “reliable cache”

Design Principles

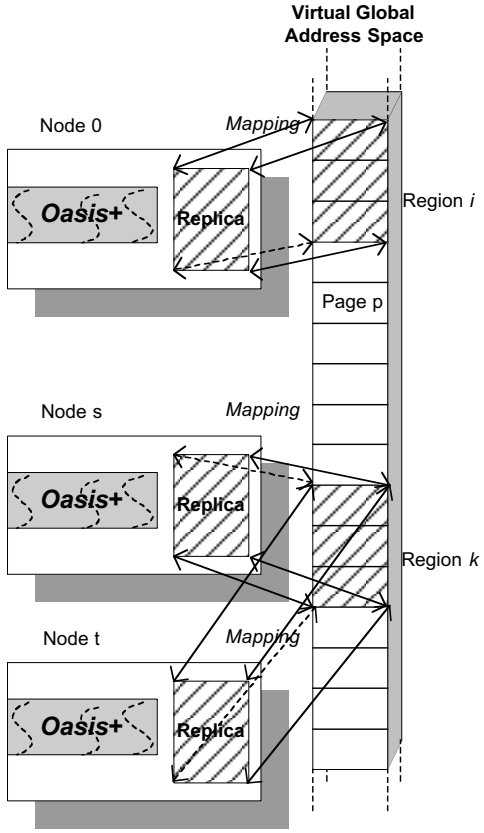
- Fault tolerance – high availability is provided through replication with atomic updates.
- High performance – eager release consistency with address range locking provides important hints to application specific memory usage.
- Cost effective – the BR (boundary restricted) coherency protocol provides configurable cost solutions.

Features of DSM

- Provides the illusion of a large a virtual address space using physical memory of each of the machines on the network
- Functionality of a cache-coherent multiprocessor
- Simple commodity computer network
- Commodity interconnect technology

Architecture

- Uses the SPREAD toolkit for reliable group services.
- I/O thread demultiplexes messages while preserving the global message order.
- Multiple worker threads maximize concurrency in multithreaded applications.



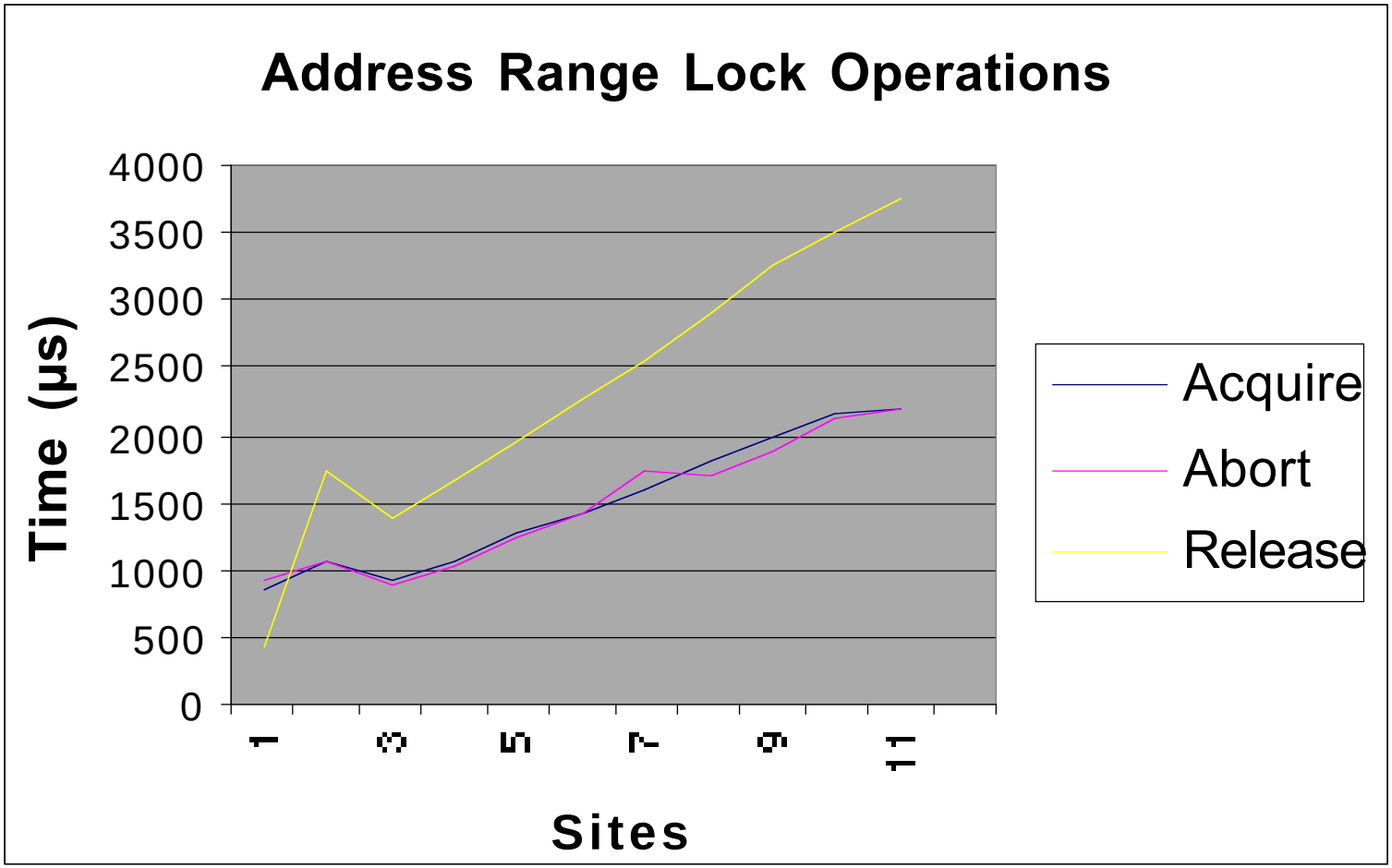
Page Management and Selection

- Distributed Page Management
  - Minimizes the burden of storage management on each site.
  - Peer to Peer implementation increases fault tolerance.
  - Shared/Exclusive locks provide low overhead mutual exclusion.
- Distributed Page Selection
  - Psuedo-Random page selection policy provides high performance by reducing the required communication.

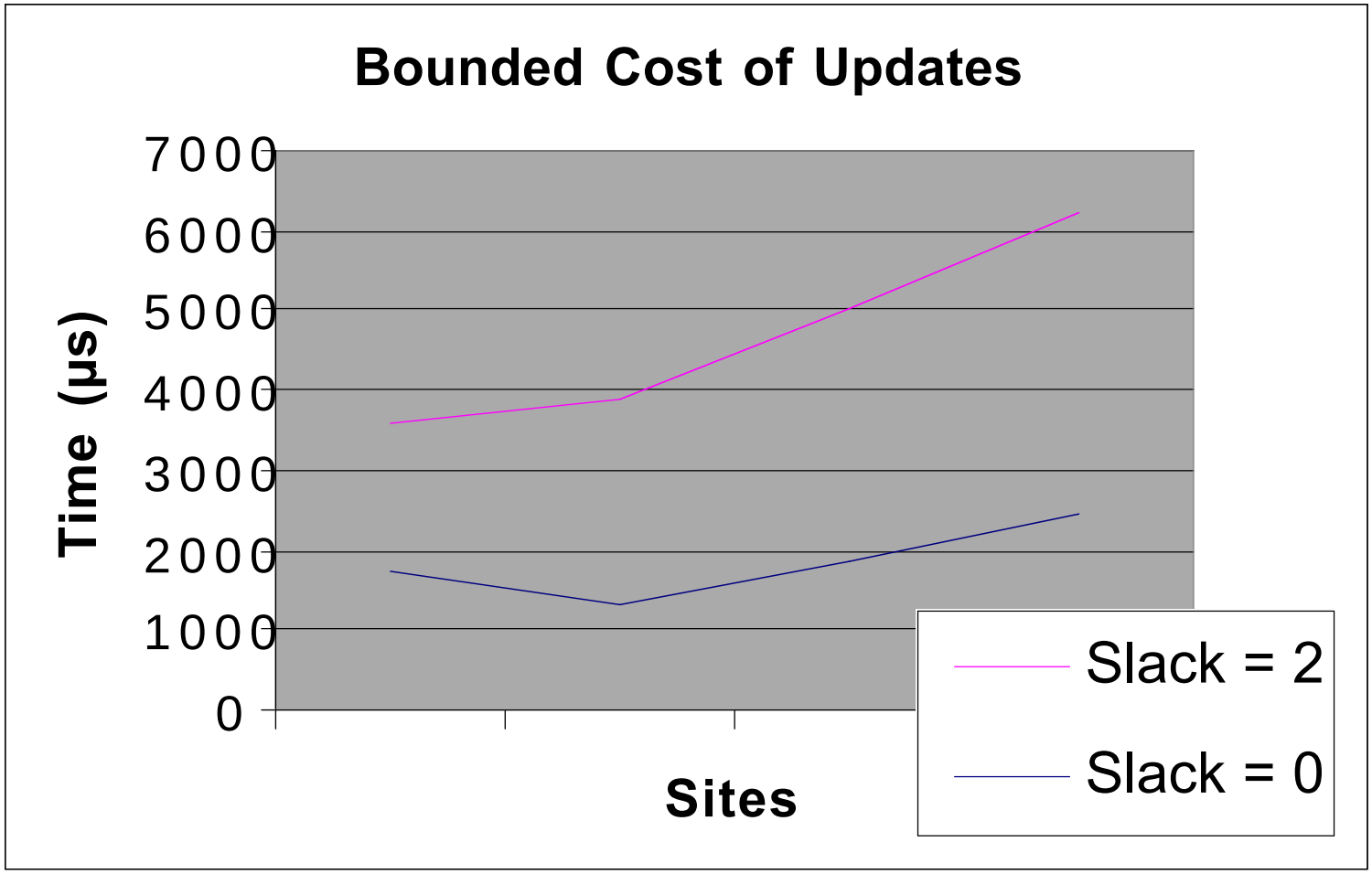
Design Features for Oasis+ Reliability

- High Availability Support with configurable replication:
  - Support for selectable degree of fault tolerance for specific DSM pages
    - Hot spots versus cold spots
  - Support for transparent failover
  - Configurable operation cost to support of only the amount of replication required
- High Performance Distributed locking that is failure resilient
- Efficient and atomic updates diffs to replicas

Performance of AR Locks



Performance of BR



Benefits and Conclusion

- Design and implementation evolving
- Our prototype uses a user-level package (instead of a kernel implementation) supporting portability, ease of debugging and more ease of maintainability
- Snap-in toolkit development techniques
- Leverage work of other research groups to minimize implementation time (object reuse)
- Move DSM from the scientific computation field to the reliability area

Future Work

- Explore alternative page selection polices.
- Examine the impact of a pre-fetch on the overall system performance.
- Compare performance with other DSM systems.