BlueSky: A Cloud-Backed File System for the Enterprise

Michael Vrable, Stefan Savage and Geoffrey M. Voelker

### Computer Science and Engineering, University of California, San Diego

Overview

Storage for small- to medium-sized organizations can be costly, both for the hardware and the management. Building storage on the cloud offers the promise of:

‣ Simpler provisioning due to elastic storage
‣ Reliability managed by cloud provider
‣ Reduced hardware requirements
‣ Simple integration with off-site backup
‣ High-performance access from multiple sites

BlueSky is our prototype for a network file system backed by cloud storage. It:

‣ Supports multiple client protocols (NFSv3 and CIFS)
‣ Supports multiple cloud storage backends (Amazon S3 and Windows Azure)
‣ Employs an on-site (for the organization) proxy mediating access and providing caching for performance and cost savings
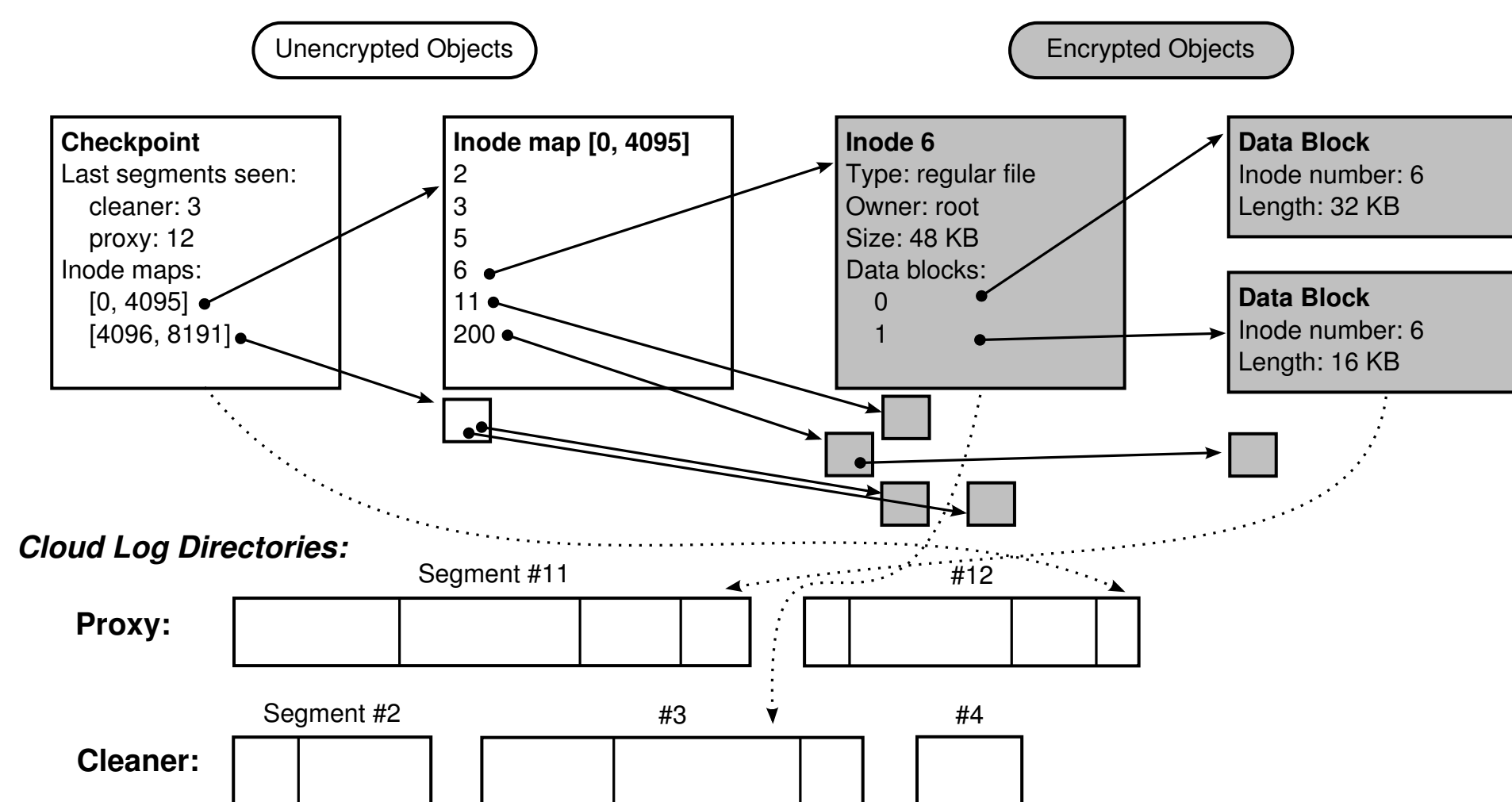‣ Explores optimizations for providing good performance at low cost

Architecture
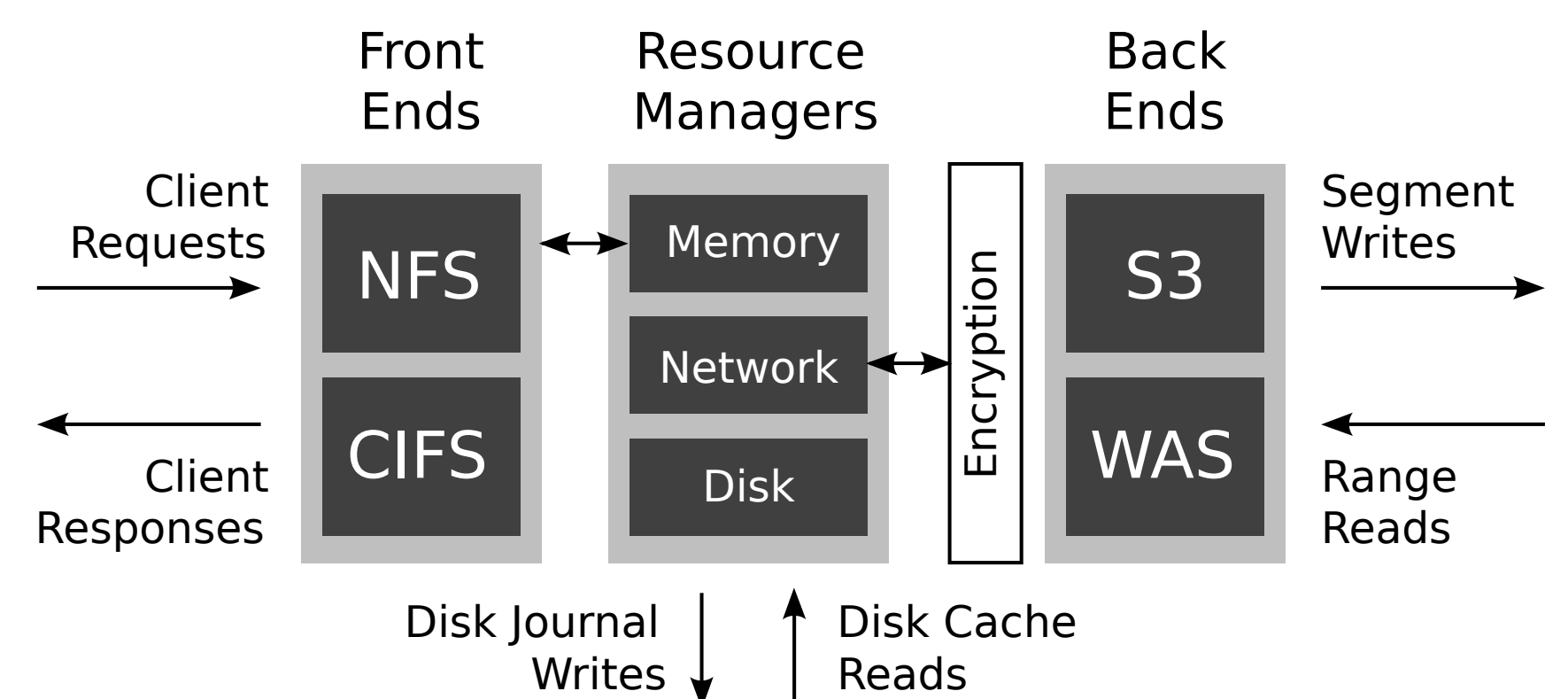
Security Goals

‣ Confidentiality: cloud cannot read file data
‣ Integrity: cloud cannot undetectably modify file system data
...but we are reliant on the cloud for availability.

File System Layout

BlueSky uses a log-structured file system layout to reduce costs and improve performance:

‣ Reduces number of operations needed to interact with storage
‣ Allows writes and reads to be batched
‣ A file system cleaner reclaims storage space
‣ Cleaner can run at the customer site or securely in the cloud for better performance/lower cost

Evaluation

‣ Comparable performance to a local Linux NFS server for workloads that largely fit in cache
‣ Significantly better performance than running network file system protocols (NFS/CIFS) over the wide area
‣ Proxy provides write-back caching to absorb write bursts without performance penalties of cloud access
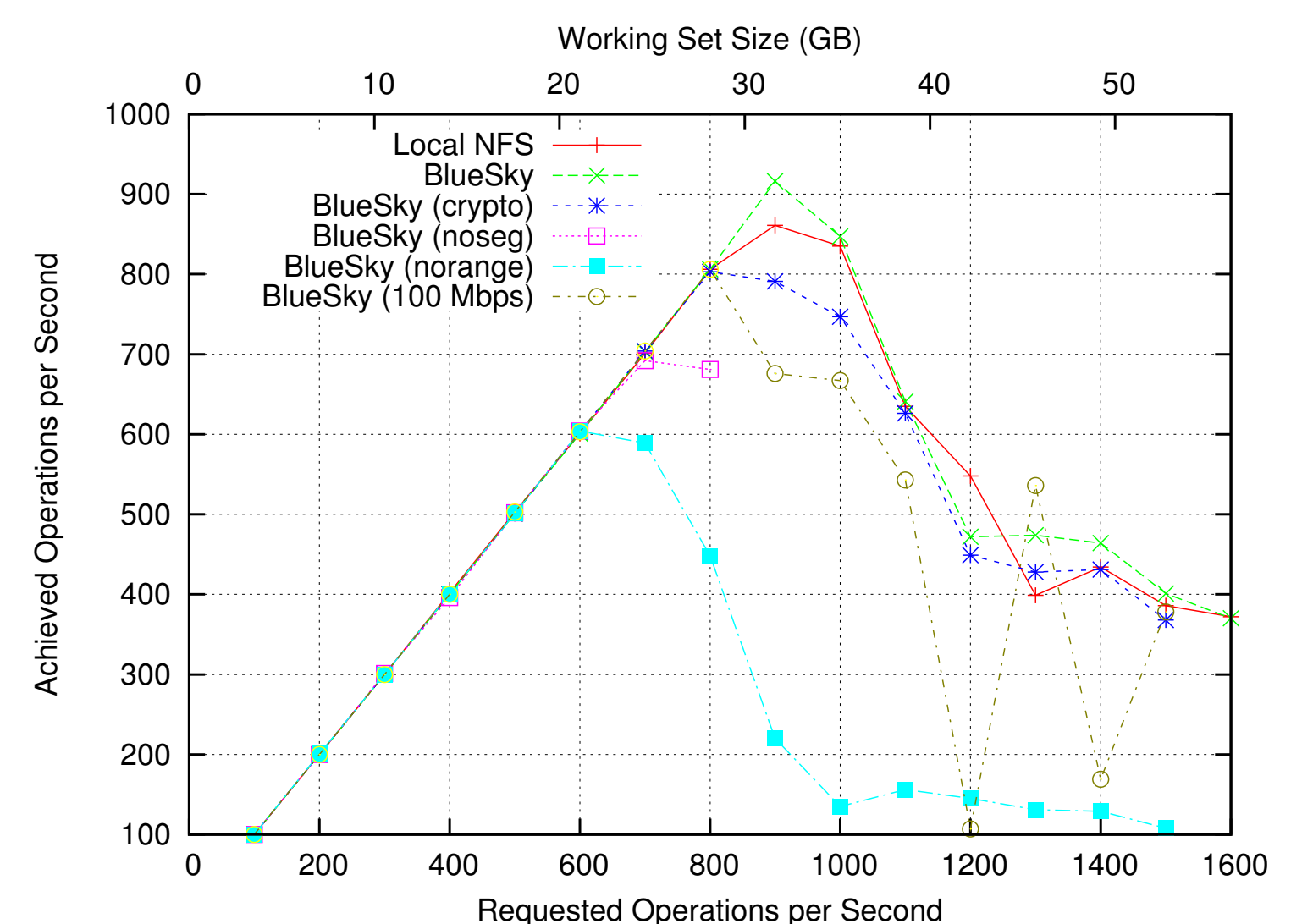
SPECsfs benchmarks show BlueSky can provide comparable-to-local-NFS performance for at least some workloads
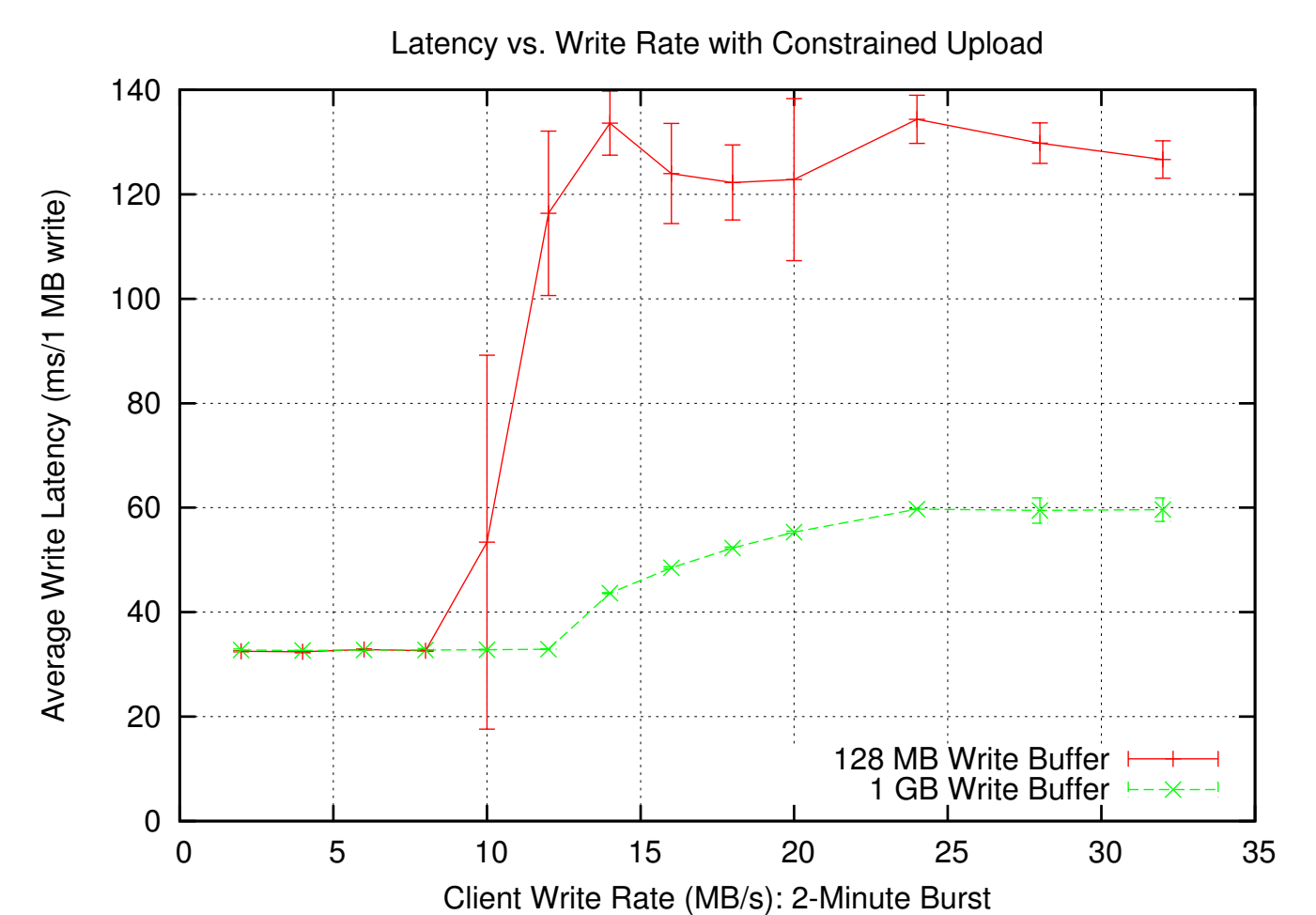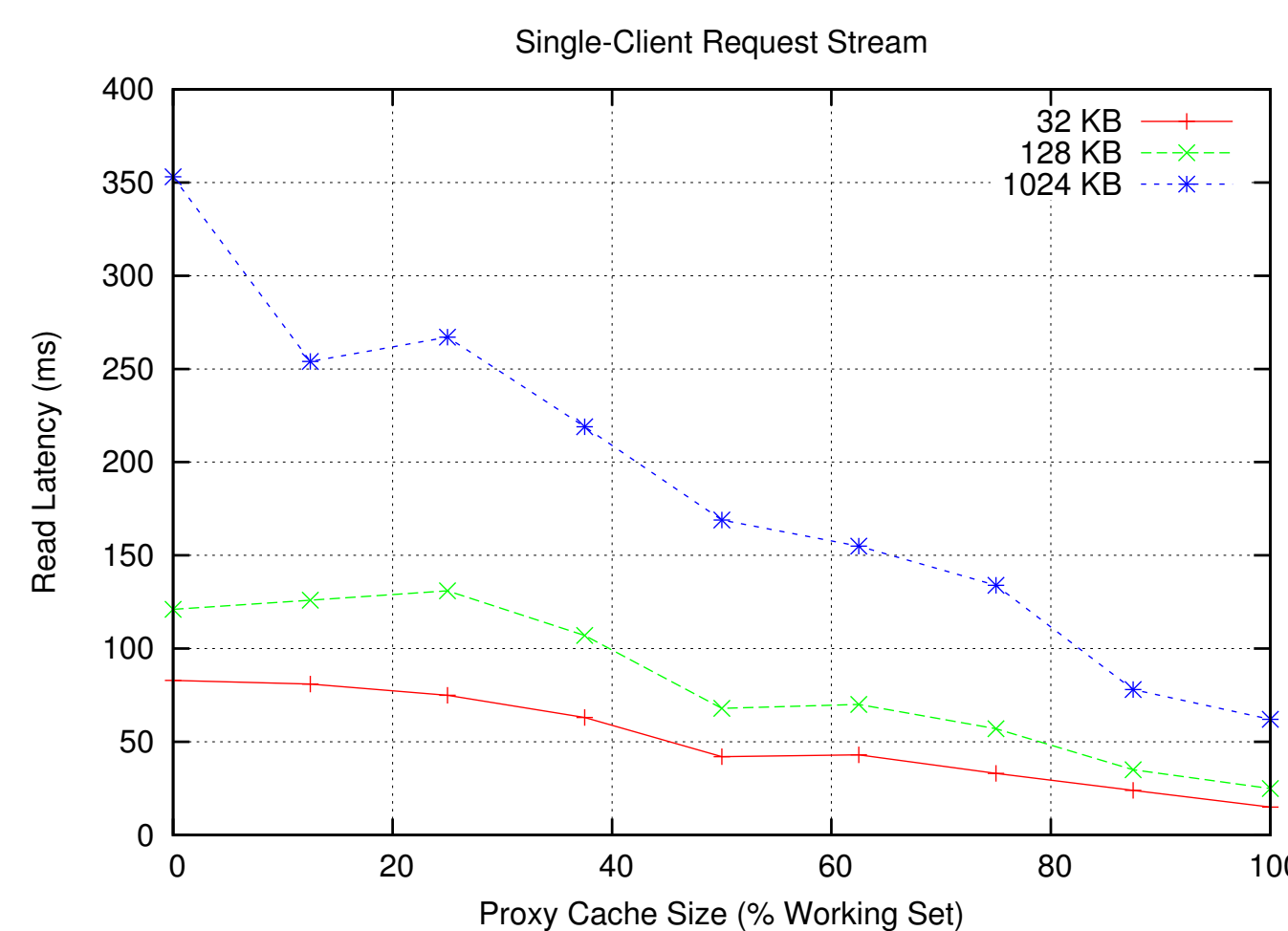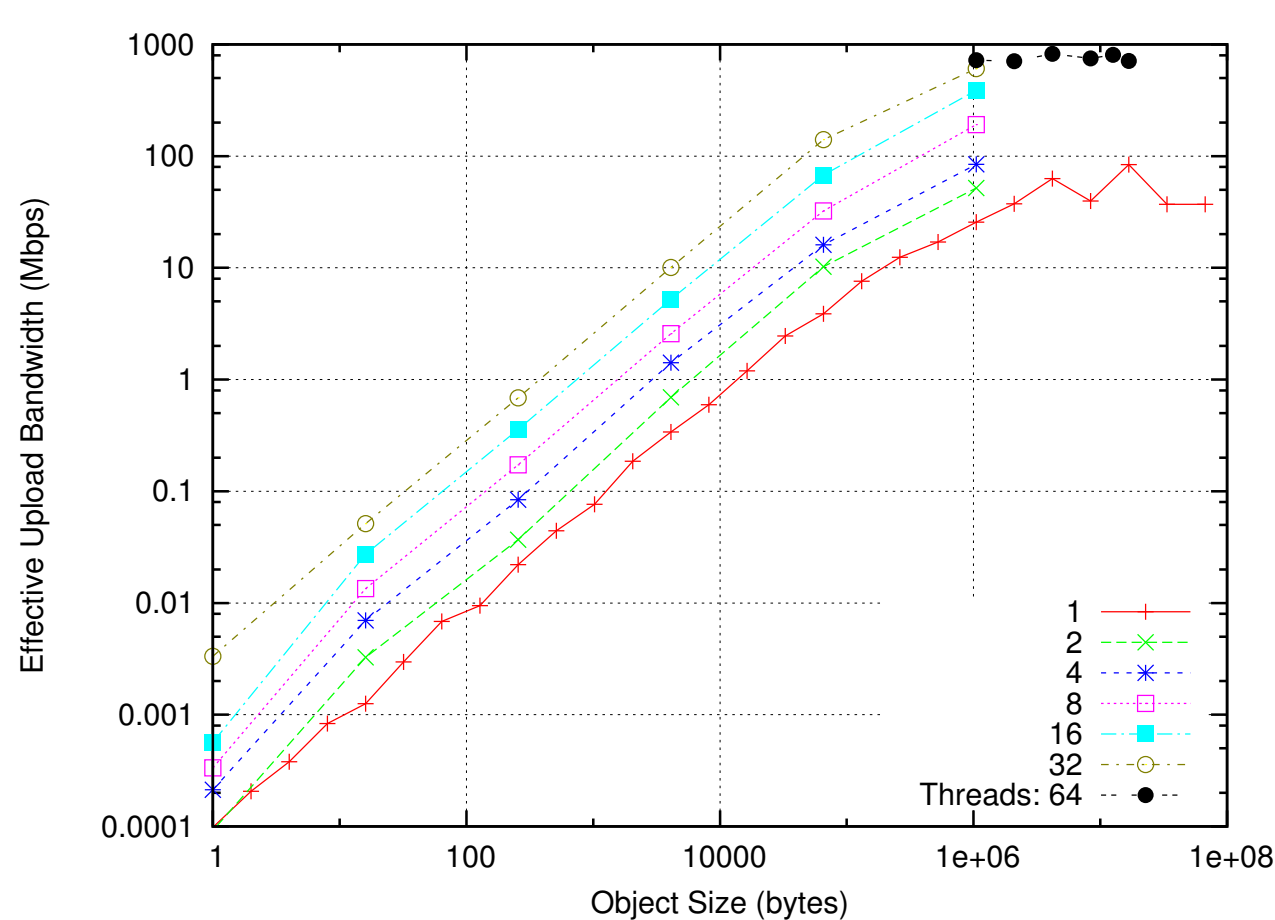
Raw performance of write access to the cloud (S3): Cloud provides very high bandwidth, but requires large writes and multiple parallel connections to achieve best performance

Read latencies as a function of effective proxy cache hit ratios: performance degrades gradually as working set/cache size ratio decreases

Write latencies depend on whether the proxy can absorb writes fully or whether data must be flushed to the cloud synchronously