Scale and Performance in a Distributed File System

Howard, Kazar, Menees, Nichols, Satyanarayanan, Sidebotham, West (1988)

What Type of Paper is This?

- Retrospective

Overview

- Give system overview.
- Define a Benchmark to measure distributed performance.
- Measure VICE-I.
- Summarize problems in VICE-I.
- Discuss VICE-II.
- Measure VICE-II.

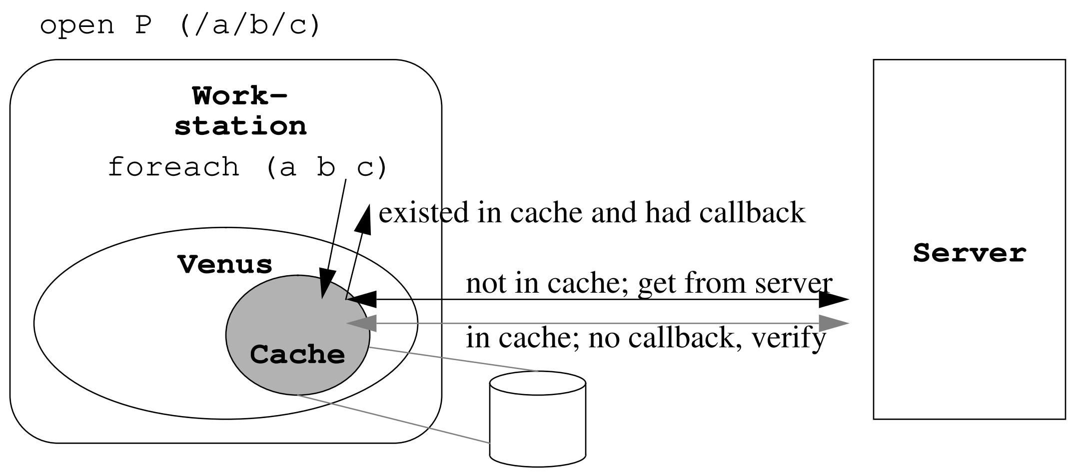
The Andrew Benchmark

- Goal: compare local and remote execution times to understand the impact of scale and distribution.
- Dataset size: 70 files; 200 KB.
- Five Phases:
    - MakeDir: Construct a target subtree.
    - Copy: Copy each file into target subtree.
    - ScanDir: Traverse hierarchy, obtaining stat information.
    - ReadAll: Read every byte.
    - Make: Compile and link the application.
- Results of Benchmark
    - Shared tree 70% slower than local tree.
    - TestAuth saturated at about 5 load units.
    - CPU utilization was peaking above 75% on servers.

- Conclusion: overall architecture is OK, but implementation could use some work.
- Use Benchmark results to motivate VICE-I to VICE-II redesign.

Major changes:

- Cache Management: callbacks.
- Naming: FIDs.
- Server Process Structure: multi-threaded process instead of per-client process.
- Low-Level File System: use access by inode calls into UNIX.



Consistency Model

- Writes are visible immediately locally; remotely in a delayed fashion.
- Upon close, writes are visible everywhere (except to existing opens).
- All other operations are globally visible.
- Workstations can operate on a file concurrently; no locking is provided.

New Performance Numbers

- Changed clients!
- Shared files only 20% slower than local.
- Scale to 20 clients with slowdown of 2X.
- Callbacks eliminate most server interaction on ScanDir and ReadAll.
- Scalability results are impressive: 70% CPU utilization at 20 load units.

Comparison with NFS

- NFS is a remote-open system (i.e. not whole-file caching).
- Run the Andrew benchmark on both systems.
- NFS time-outs improperly handled by applications, result in errors.
- The results they show demonstrate AFS is superior to NFS except at very low load.
- Andrew claims superior scalability.

Operability

- Volumes: small groupings of files.
- Map volumes to users
- Multiple volumes to a disk partition.
- Can move volumes just by updating volume database.
- Move volumes by creating clones, moving clone, repeating until there are no more updates.
- Quotas enforced per volume.
- Backups handled via clones.