Distributed File System Support for Virtual Machines in Grid Computing

Dongyun Jin (djin3@uiuc.edu)

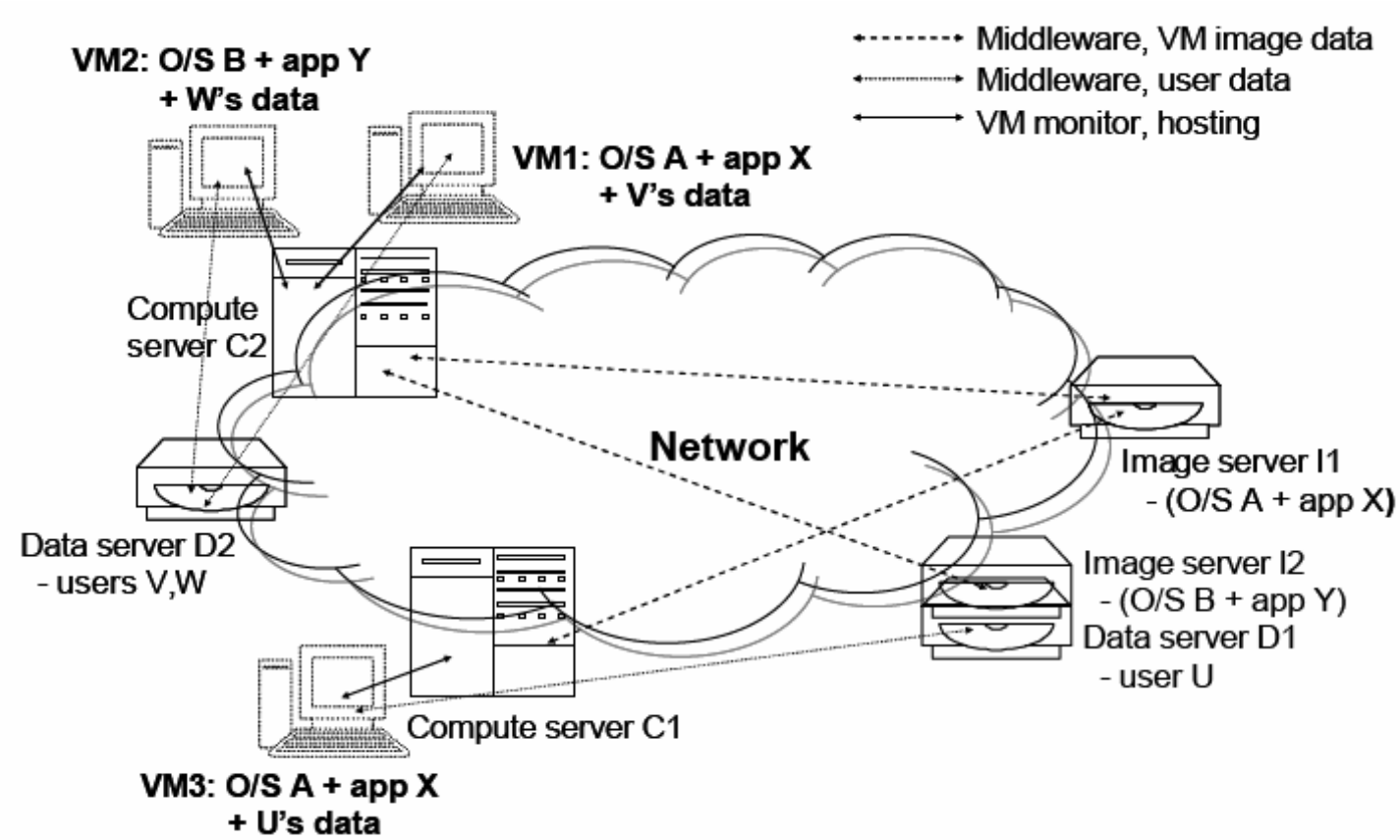
CS523

April, 2006

Outline

- Overview of Grid Computing using VM
- Goals
- Previous Works
- Approaches
  - On-demand access
  - User-level and write-back disk caches
  - Middleware-driven consistency
  - Use of meta-data
- Performance
- Discussion & Reference

Grid Computing using VM



Goals

- Distributed Grid Virtual File System
  - Without modifications to existing NFS clients and servers.
  - Support the execution of unmodified application binaries
  - Not specific to a particular VM tech.
  - Efficient instantiation of VMs across dist domains

Previous Works

- Current Grid data management solutions typically employ file-staging techniques
  - Require the user to explicitly specify the files that need to be transferred. (GridFTP)
  - Transfer entire files. (GASS)
- DM solns supporting on-demand transfer for Grids require customized application libraries and/or file servers.

Approaches

- Distributed Virtual File System
  - Virtualization layer on top of NFS
- On demand access
- User-level and write-back disk caching
- Write-back policy
  - Middleware-driven consistency
- Using of Meta-data

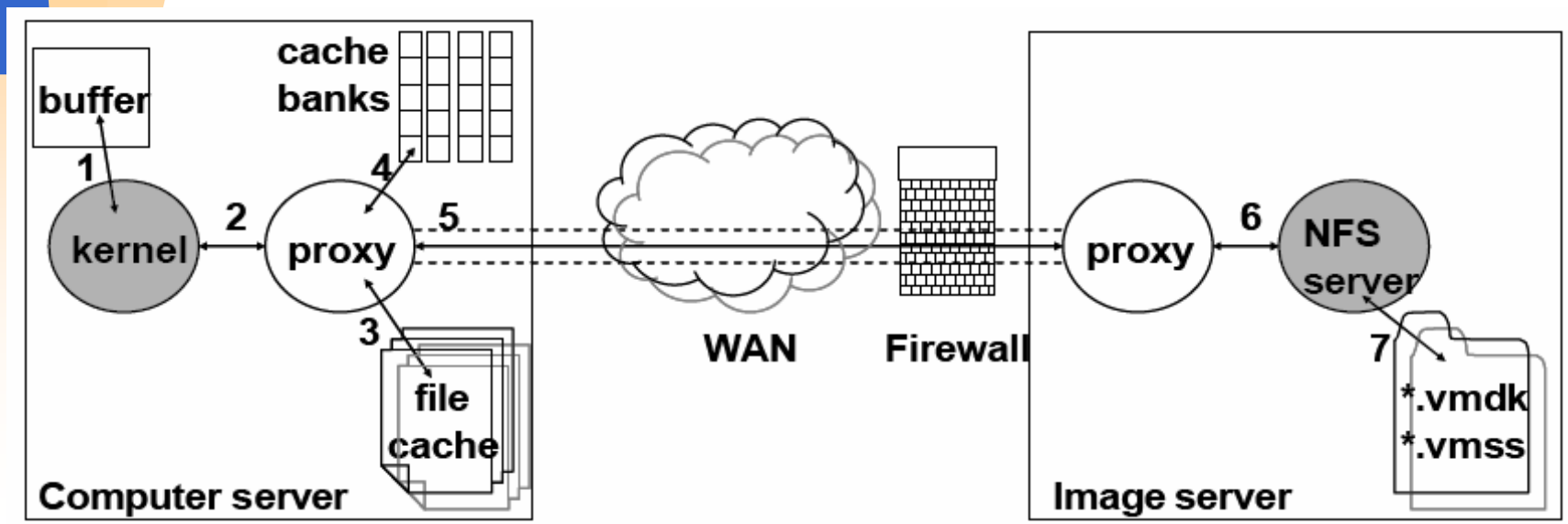
On demand access

- Distributed Virtual File System can be built via extensions to existing NFS implementations that are at user-level.
  - On-demand between Grid storage and compute servers.
  - Requiring neither modification of O/S clients and servers, nor of applications.

User-level and write-back disk caching

- Disk-based file system caches
  - AFS transfers and caches entire files in the client disk.
  - CacheFS supports disk-based caching.
  - -> these designs require kernel support!!!
- GVFS is extended to employ client-side proxy managed disk cache.
- GVFS proxy disk cache supports the write-back policy to hide long write latency

User-level and write-back disk caching



Middleware-driven consistency

- NFS clients are not aware of the existence of other potential sharing clients.
  - Maintaining consistency is difficult.
- It supports O/S signals for middleware-controlled writing back and flushing of cache contents.
  - Middleware-driven consistency

Using of meta-data

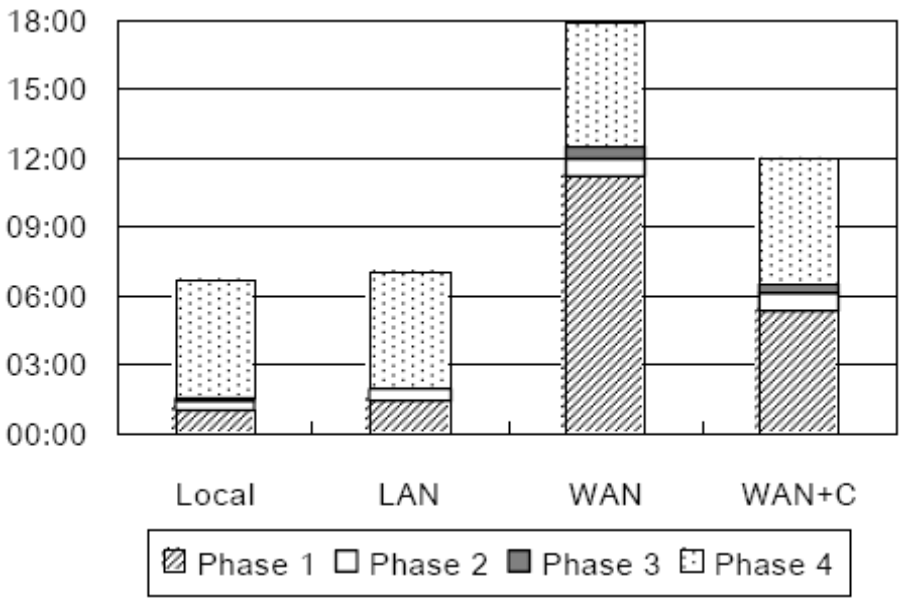
- Memory state file
  - Meta-data file specify which blocks in the memory state are all zeros.
- Whether it require entire file or part of it.

Performance

- Lan Image server
  - Dual 1.8GHz Pentium 3 Cluster node
  - 1GB RAM, 576GB Harddisk
- Wan Image server
  - Dual 1GHz Pentium 3 Cluster node
  - 1GB RAM, 45GB Harddisk
- Compute server(For app)
  - 1.1GHz Pentium 3 Cluster node
  - 1GB RAM, 18GB SCSI Harddisk
- Compute server(For VM cloning)
  - Four 2.4GHz Xeon
  - 1GB RAM, 18GB SCSI Harddisk
- 100Mbps for LAN

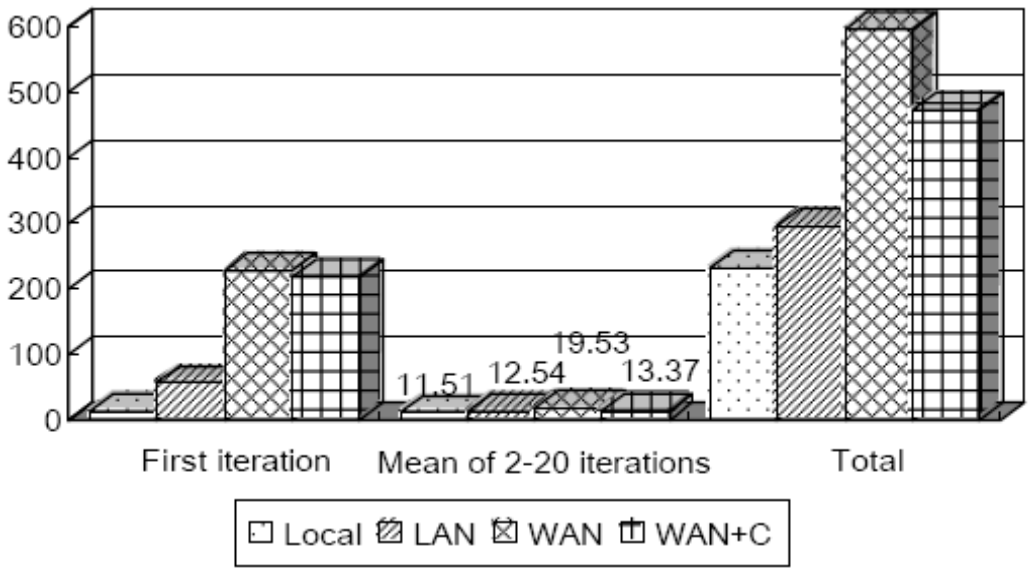
Performance

- SPECseis96
  - Compute-intensive part is within a 10% range
  - Phase 1 shows a large difference



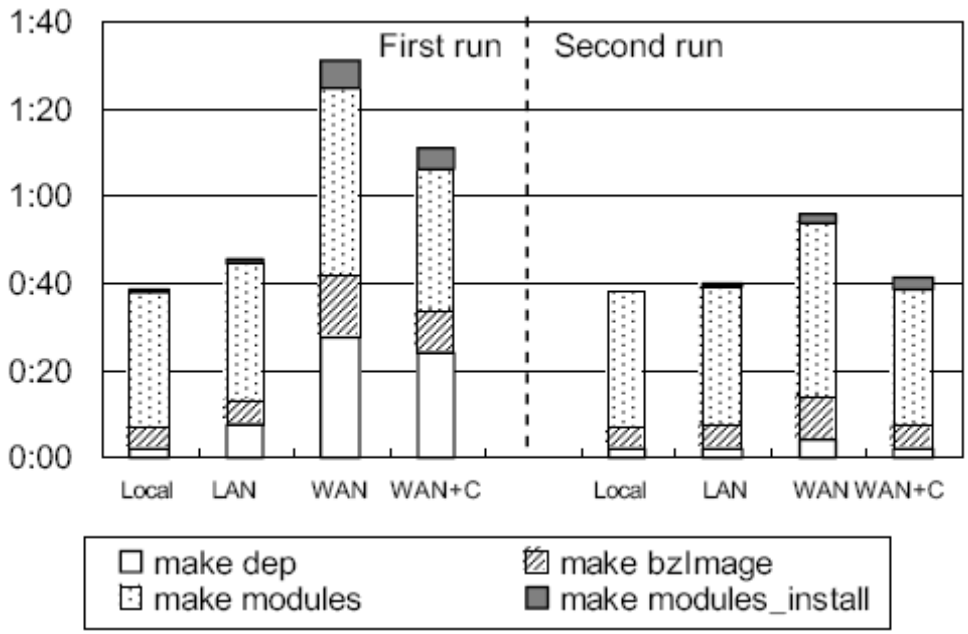
Performance

- LaTex



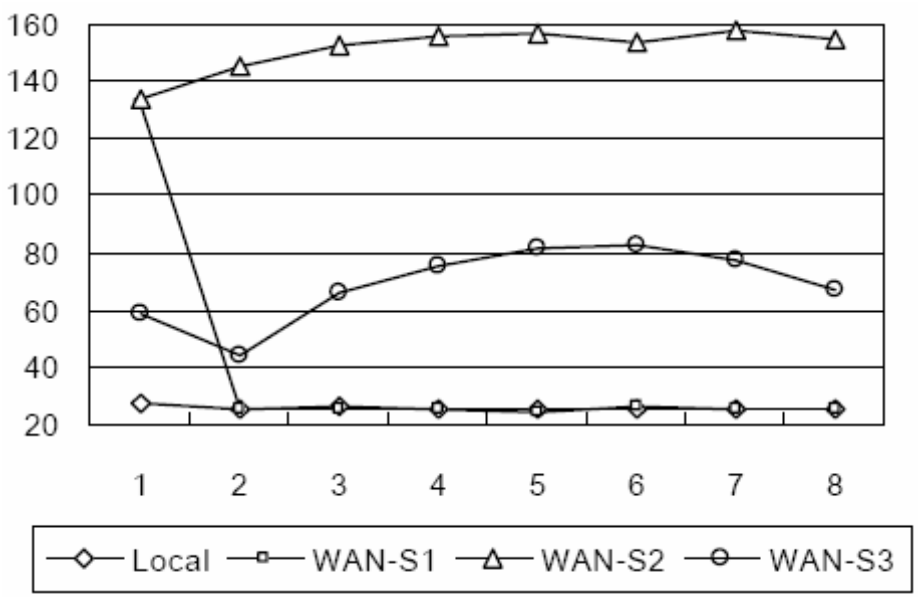
Performance

- Kernel Compilation



Performance

- Virtual Machine Cloning



Conclusion

- For a highly flexible grid computing environment, it is important that Grid middleware provides efficient data management service for VMs.
- Results show that user-level proxy caches improve upon the performance of conventional NFS over a WAN.

Future Work

- Support for efficient checkpointing
- Migration of VM instances for load-balancing and fault-tolerant execution.
- Dynamic profiling of application data access behavior.

Discussion and References

- Any Question?

- M. Zhao, J. Zhang, and R. Figueiredo. “Distributed File System Support for Virtual Machines in Grid Computing”. In *HPDC 2004*, 2004.
- B. Pawlowski, C. Juszczak, P. Staubach, C. Smith, D. Lebel and D. Hitz, “NFS Version 3 Design and Implementation”, Proc. USENIX Summer Technical Conference, 1994