A User-level Secure Grid File System

Ming Zhao, Renato J. Figueiredo

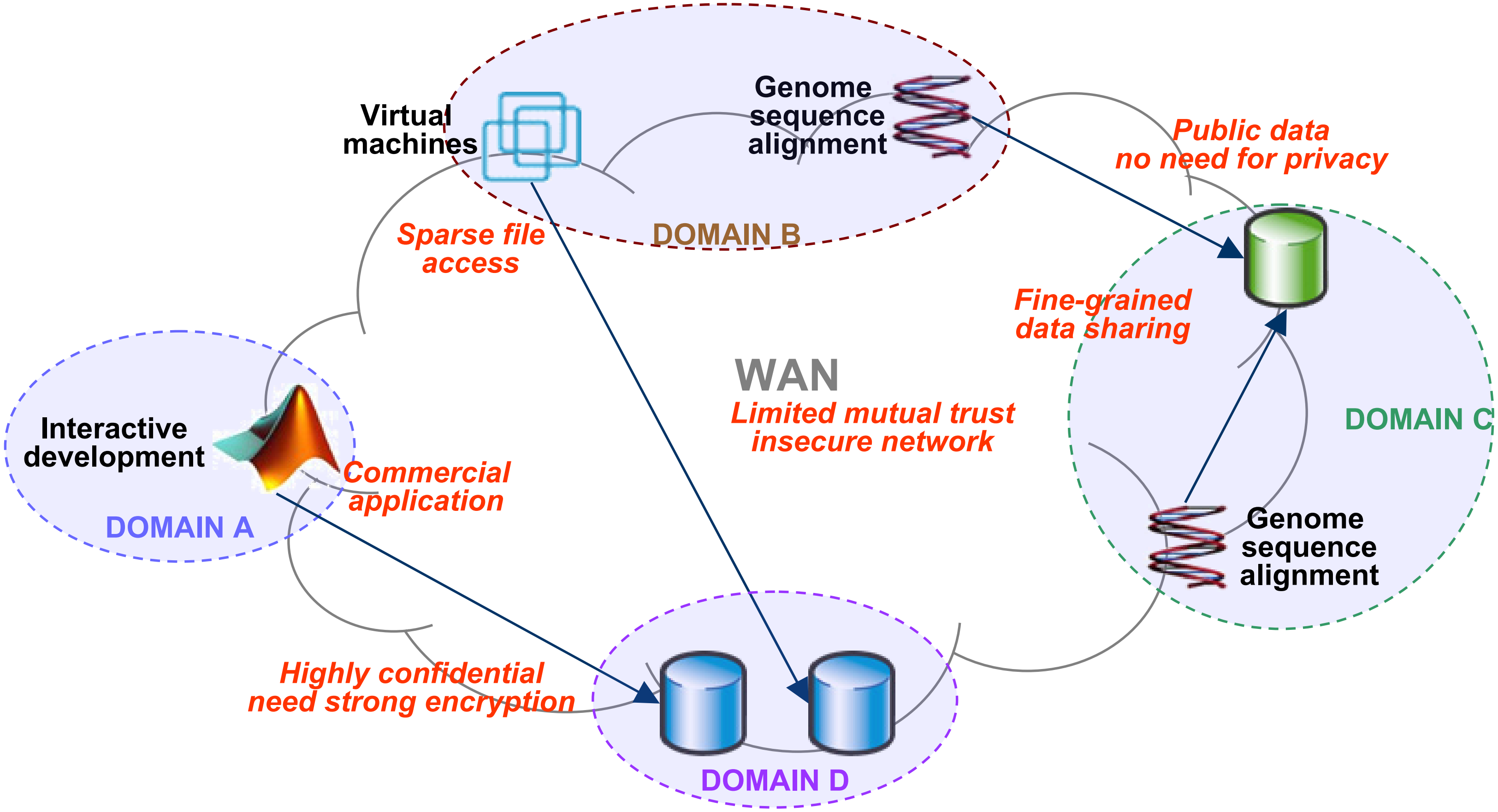
Advanced Computing and Information Systems (ACIS)
Electrical and Computer Engineering

University of Florida

{ming, renato}@acis.ufl.edu

Motivations

- Need for secure grid file systems
  - Support for unmodified applications, fine-grained data sharing
  - Support for *strong*, *flexible* and *grid-compatible* security


Virtual machines
Genome sequence alignment
Public data no need for privacy
Sparse file access
DOMAIN B
Fine-grained data sharing
WAN
Limited mutual trust insecure network
DOMAIN C
Interactive development
Commercial application
DOMAIN A
Genome sequence alignment
Highly confidential need strong encryption
DOMAIN D

ACIS

Overview

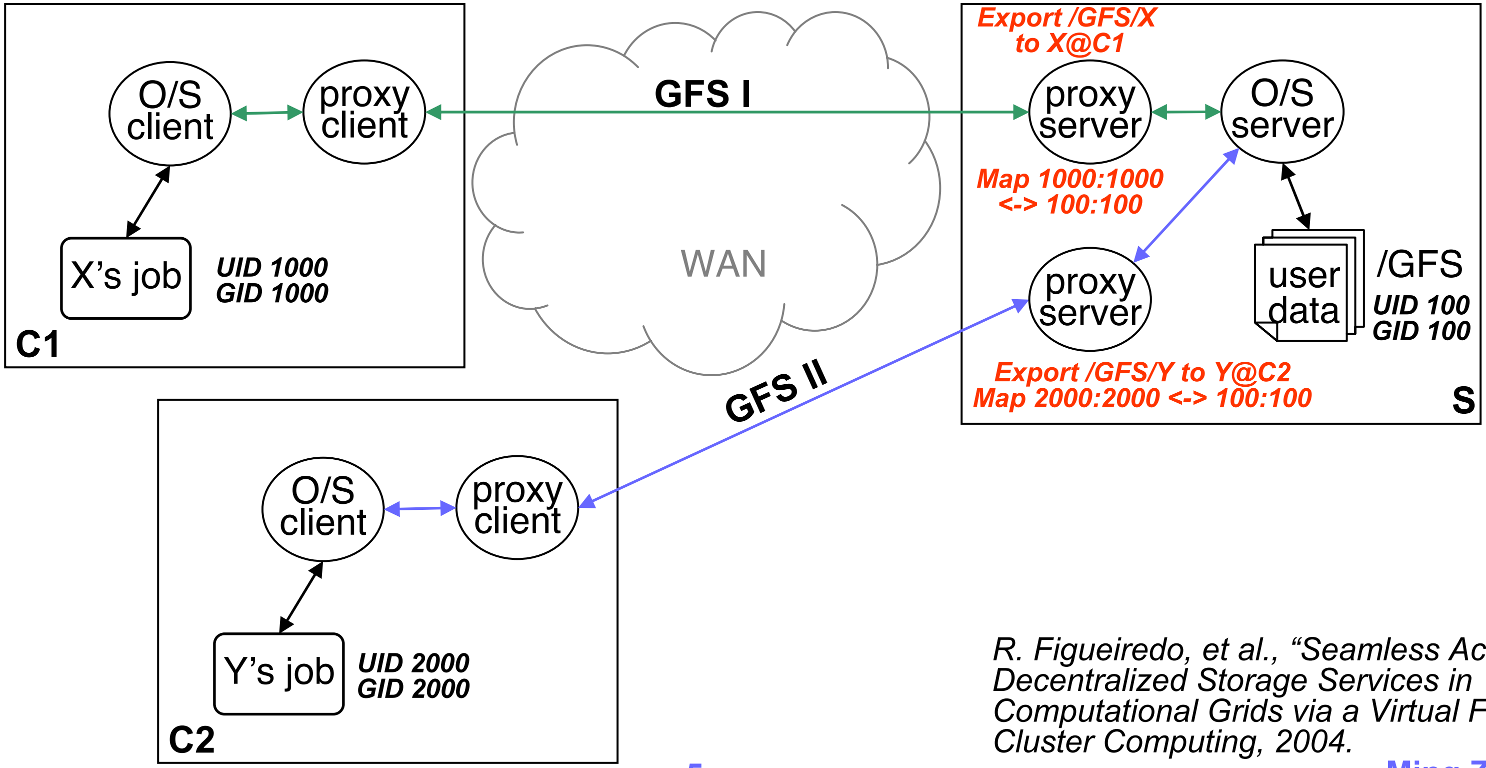
- Goal
  - Secure DFS-based grid data management
- Approach
  - A user-level secure grid file system
- Contributions:
  - Secure and efficient grid data access
  - Secure services for management and configuration
  - Support for unmodified applications and O/Ss
  - Flexible configurations based on application needs
  - Compatible with widely-accepted grid security infrastructure

Outline

- Background
- Architecture
  - Secure GFS-based data access
  - Secure service-based management
- Implementation
  - Secure Remote Procedure Calls
  - GSI-based GFS proxy
  - Grid file access control
  - GSI-based management services
- Performance
- Summary

Background

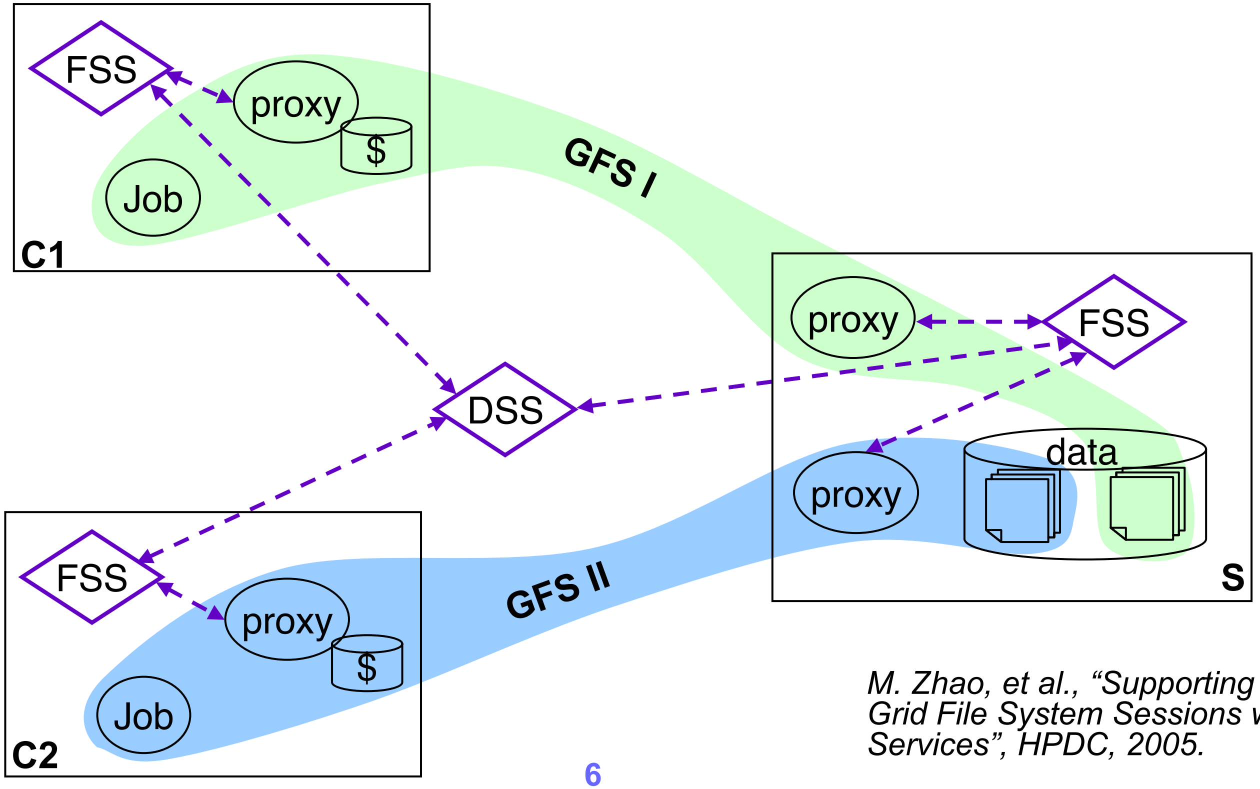
- Grid File System (GFS, a.k.a. GVFS)
  - User-level virtualization of distributed file systems via proxies
  - Leverages widely deployed O/S clients and servers (NFS V3)
  - Proxies control authentication, authorization, identity mapping
  - Per session security configuration and enforcement



R. Figueiredo, et al., "Seamless Access to Decentralized Storage Services in Computational Grids via a Virtual File System", Cluster Computing, 2004.

Background

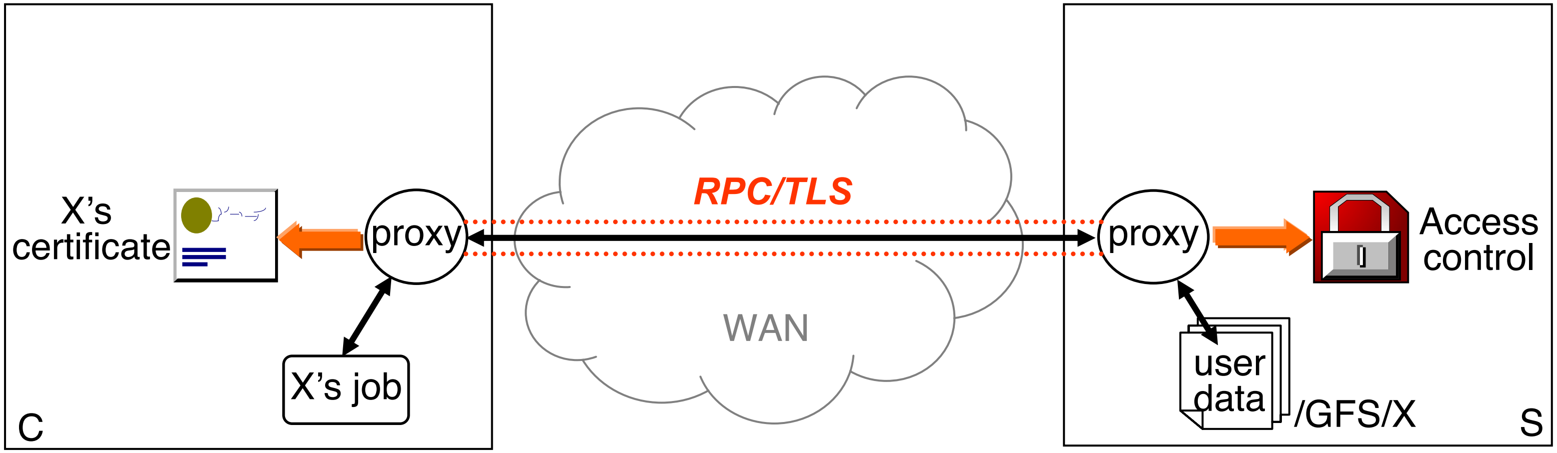
- Data Management Services
  - Middleware for controlling the lifecycles and configurations of GFSs
  - File System Service (FSS)
    - Controls local proxies to establish and configure GFSs
  - Data Scheduler Service (DSS)
    - Schedules and customizes GFSs through interactions with FSSs



M. Zhao, et al., "Supporting Application-Tailored Grid File System Sessions with WSRF-Based Services", HPDC, 2005.

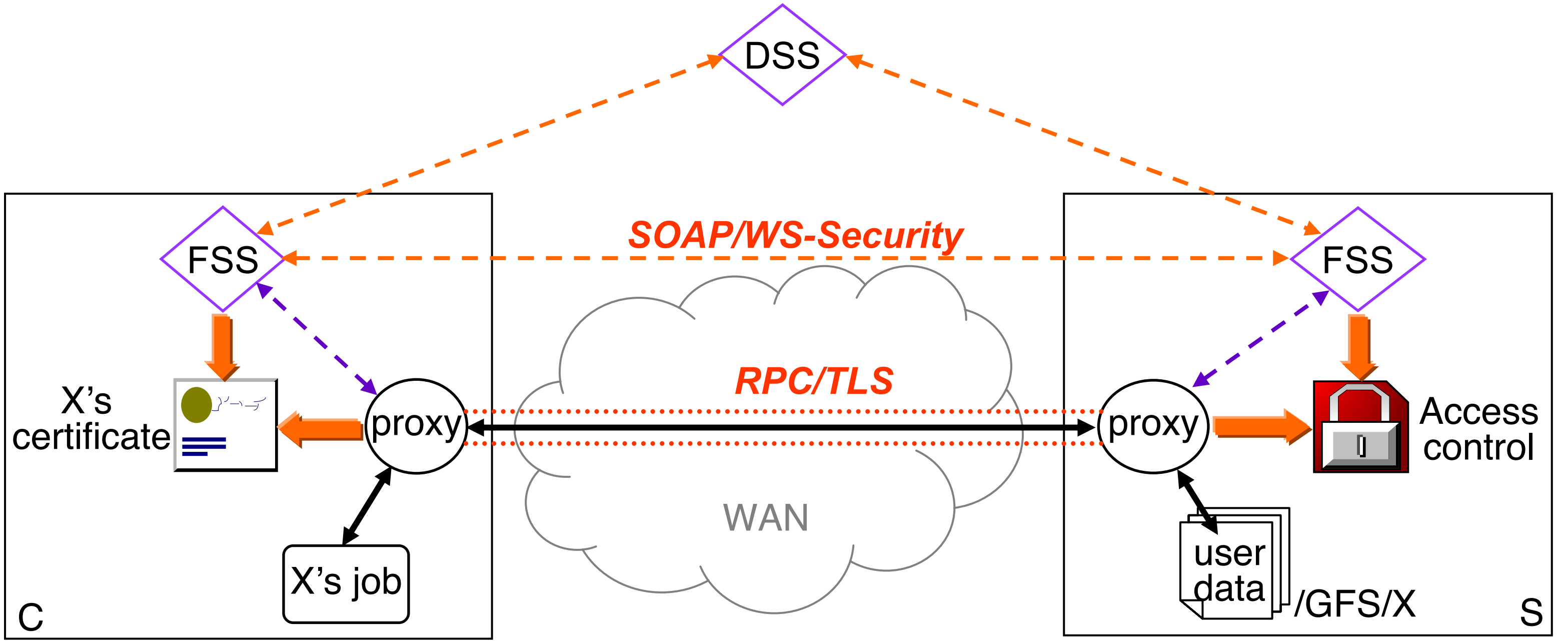
Secure GFS-based Data Access

- Based on Transport Level Security (TLS)
  - Efficient end-to-end secure channel for remote procedure calls (RPC)
  - Grid user's (proxy) certificate is used for authentication and authorization
  - Shared key is negotiated for encryption of GFS traffic
  - Digital signature or MAC is used for integrity checking
  - Grid-style ACL associates file access permissions with grid user identity
  - Flexible and customizable security policies and mechanisms per GFS



Secure Service-based Management

- Based on Message Level Security (MLS)
  - Protection of messages in service-level interactions (SOAP)
  - Support for security cooperation with other middleware services
  - Grid user or service authenticates with DSS using the user's certificate
  - Authorization is done by checking an ACL or a dedicated service
  - FSS controls proxy client to use the user's certificate to set up GFS



Secure RPC

- Based on Socket Layer Security (SSL/TLS)
  - Efficient implementations and successful deployments
  - Support for full-featured security and a wide range of algorithms
  - Transparent protection of GFS traffic
    - GFSs are set up on per-user/application basis

- SSL-enabled secure RPC library (SRPC)
  - We have developed it based on TI-RPC and OpenSSL
  - API examples
    - *clnt_tli_ssl_create(... ... , struct security_context)*
    - *svc_tli_ssl_create(... ... , struct security_context)*
  - Security configurations are defined in the *security_context* struct
  - Generic secure RPC support, no need for system-level changes

GSI-based File System Proxy

- Enhancements
  - Uses SRPC library for secure communications
  - Parses and validates GSI (Grid Security Infrastructure) certificates for authentication and authorization
- Configurations
  - Defined in a configuration file used by users or services
    - Security policies, algorithms, and parameters
  - Support for dynamic reconfigurations
    - Change of security policies
    - Reload of certificates
    - Renegotiation of session keys

Grid File Access Control

- Per-GFS gridmap file
  - Per file system access control
  - Maps grid user identities to local user accounts
  - A grid users gains the same file access permissions as the mapped local user

- Per-file/directory ACL file
  - Per file/directory access control
  - Stored as a protected hidden file: *.filename.acl*
  - Contains pairs of grid user identity and access permission bitmask
  - Leverages NFS ACCESS RPCs for checking ACL files and returning access permission bitmasks

GSI-based Management Services

- Based on Web service standards
  - Services based on WSRF (Web Service Resource Framework)
    - Managing GFS states as resources
  - Service-level security based on WS-Security
    - Signing and verifying SOAP messages
  - Implemented with WSRF::Lite
  - Interoperable with other middleware services

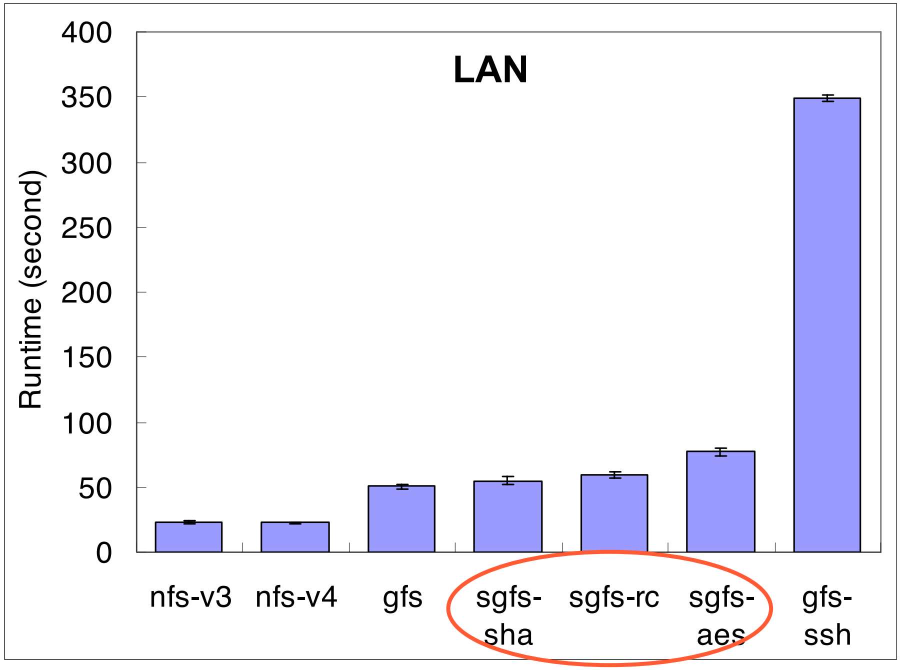
- Grid file access control
  - Uses ACLs stored in database
    - Creates gridmap or ACL files for proxies
  - Leverages dedicated security services
    - E.g. Community authorization service

Experimental Setup

- File system clients and servers
  - Virtual machines
    - Hosted on cluster nodes (3.2GHz hyperthreaded Xeon CPUs, 4GB memory)
- Network
  - LAN
    - Gigabit Ethernet
  - WAN
    - Emulated with NIST Net
- Benchmarks
  - File system benchmarks
    - IOzone, Postmark
  - Applications
    - Software development, scientific computing

IOzone

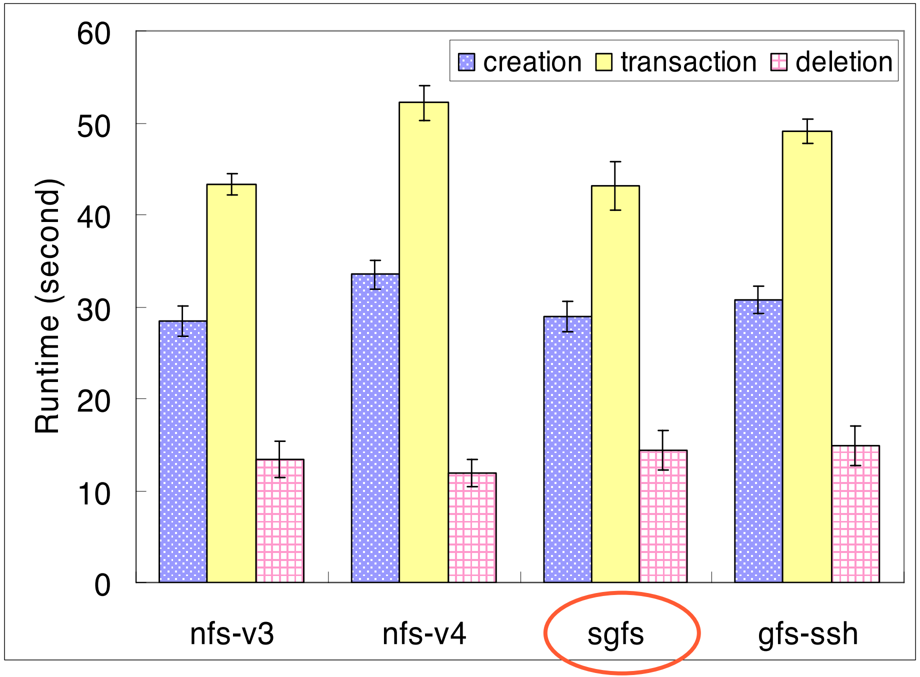
- Intensive sequential reads
  - LAN
  - No client-side caching, no server-side disk accesses
- Configurations
  - *NFS V3/V4*
    - Native, unsecured NFS
  - *GFS*
    - Unsecured GFS
  - *GFS-SSH*
    - SSH tunneling of GFS
  - *SGFS*
    - *sgfs-aes*: **AES-256bit, SHA1-HMAC**
    - *sgfs-rc:* **RC4-128bit, SHA1-HMAC**
    - *sgfs-sha:* **SHA1-HMAC**



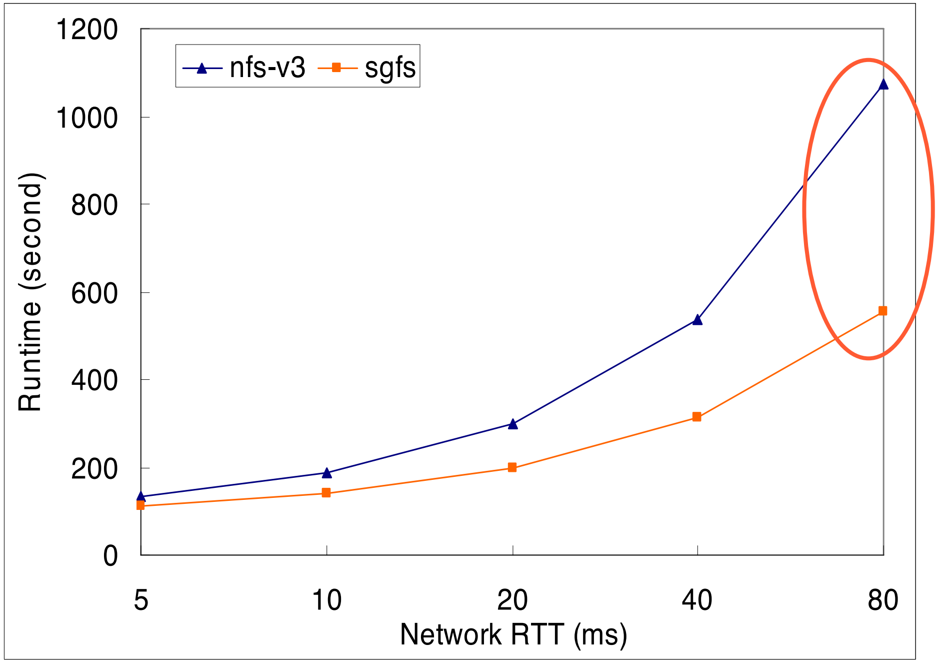
- Performance
  - 2-fold overhead for user-level security
  - Stronger security sacrifices more performance
- *sgfs-aes* is used for the following study

Postmark

- Simulates workloads from emails, news, Web commerce
  - *Creation*, *transaction*, *deletion*
  - Intensive small reads/writes and metadata updates
- Performance
  - GFS outperforms native NFS (by using aggressive attributes caching)
  - Speedup increases as network latency grows



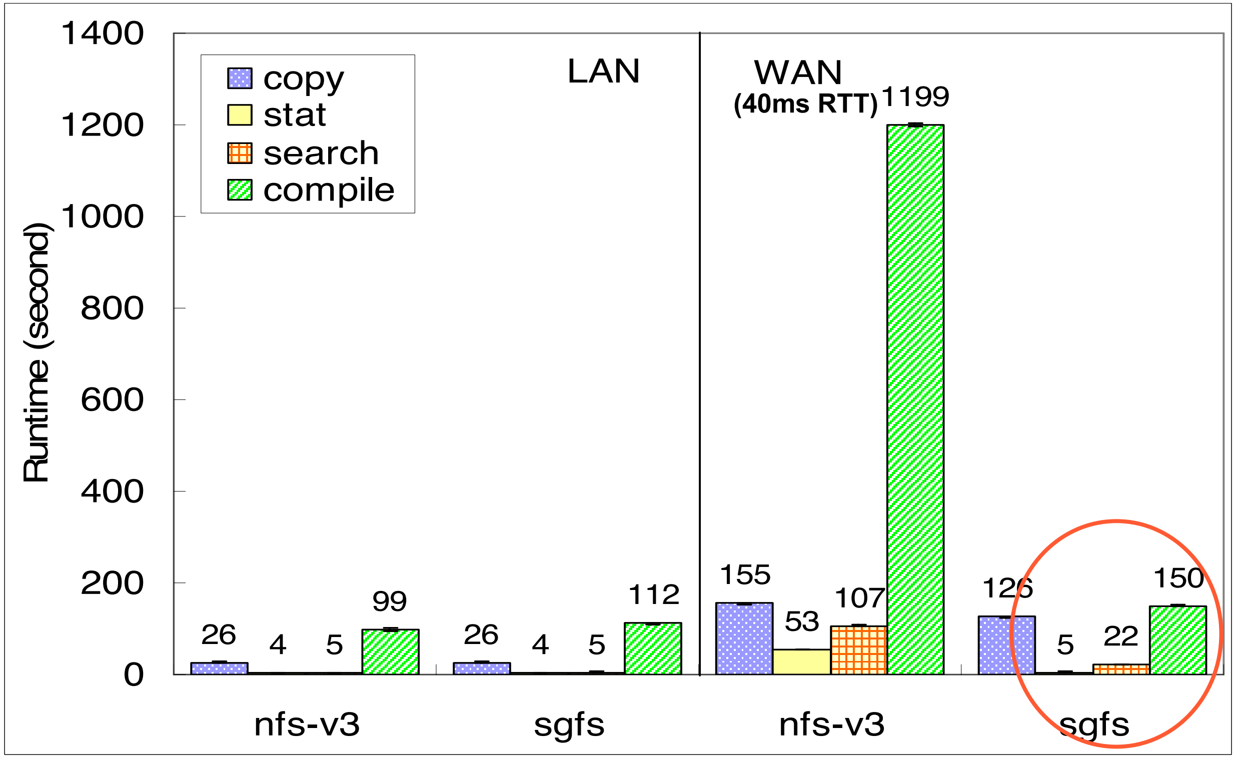
LAN



WAN

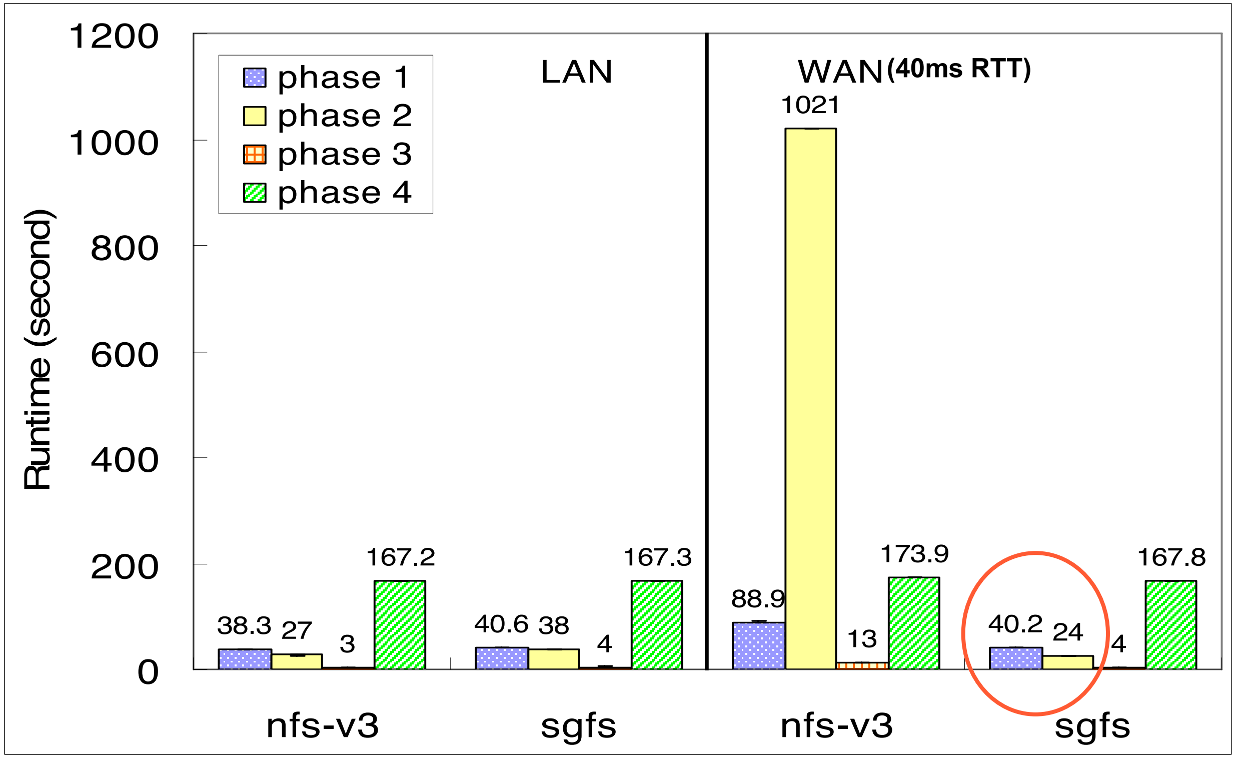
Modified Andrew Benchmark

- Models software development process
  - *Copy*, *stat*, *search*, *compile*
  - Uses a larger workload than the original Andrew benchmark
- Performance
  - Very close to native NFS on LAN
  - Significant speedups on WAN (by using disk caching for attributes and data)



Seismic

- Models computing and data intensive scientific applications
  - *Phase 1*: generate a large output file
  - *Phase 2, 3, 4*: process data
- Performance
  - Very close to native NFS on LAN
  - Significant speedups on WAN (by using disk caching with write-back)



Related Work

- Security in distributed file systems
  - NFS (V2, V3)
  - NFS (V4), GridNFS
  - AFS (OpenAFS, Coda)
  - SGFS supports unmodified O/Ss, strong security for grid data access, and flexible application-tailored configurations
- Security in grid data management
  - Globus
  - Legion
  - Condor
  - SGFS combines the advantages of TLS and MLS, and is compatible with existing grid systems based on GSI

Summary

- Problem
  - Secure DFSs for grid data management
- Solution
  - A user-level secure grid file system
    - Strong and compatible security for grid data access
    - Seamless support and flexible customization for applications
    - Convenient integration with grid resources and systems
- Future work
  - User-level cryptographic functions for protection of data storage

Acknowledgments

- In-VIGO team
  - http://invigo.acis.ufl.edu
- Sponsors
  - NSF
  - IBM SUR
  - DURIP
- **Questions?**
  - ming@acis.ufl.edu
  - http://www.acis.ufl.edu/~ming