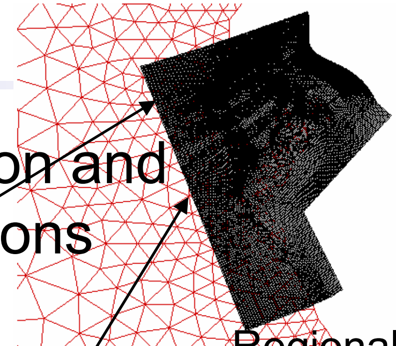
Supporting Application-Tailored Grid File System Sessions with WSRF-Based Services

Ming Zhao, Vineet Chadha, Renato Figueiredo

Advanced Computing and Information Systems
Electrical and Computer Engineering
University of Florida

Motivating Example

- Shared file system facilitates communication and synchronization between coupled applications



Regional-scale, curvilinear simulation grid of Tampa Bay

Every 30 timesteps
1.5MB per exchange

Curvilinear-grid **H**ydrodynamics **3D** model

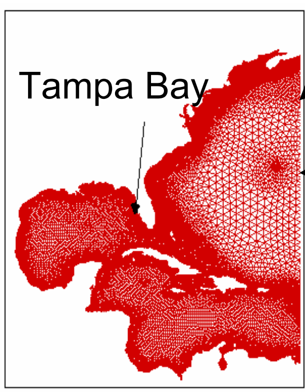
CH3D

SWAN

Simulating **WA**ves **N**earshore model

Every 30 timesteps
1.8MB per exchange

Every timesteps
40KB per exchange



ADCIRC

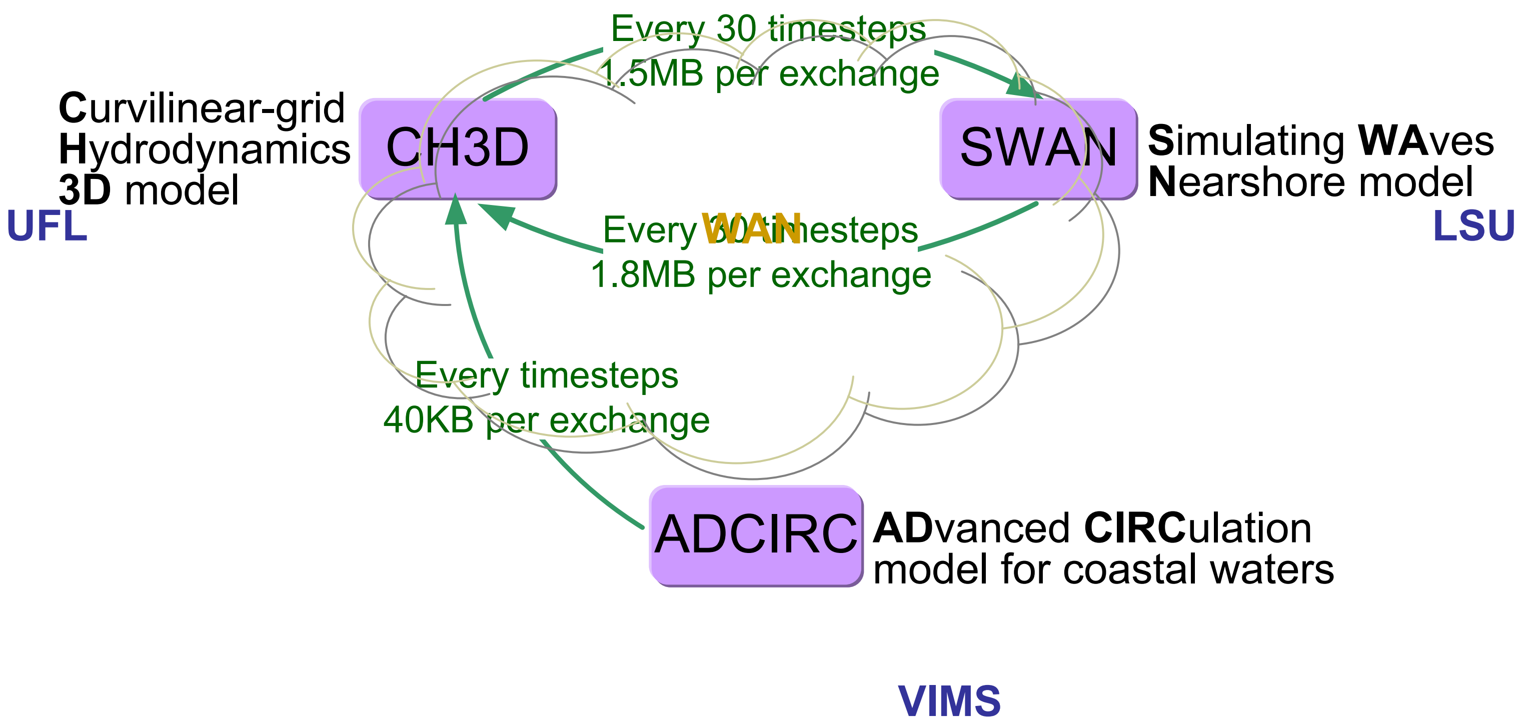
ADvanced **CIRC**ulation model for coastal waters

Basin-scale, unstructured ADCIRC simulation grid

Coastal surge coupled modeling

Motivating Example

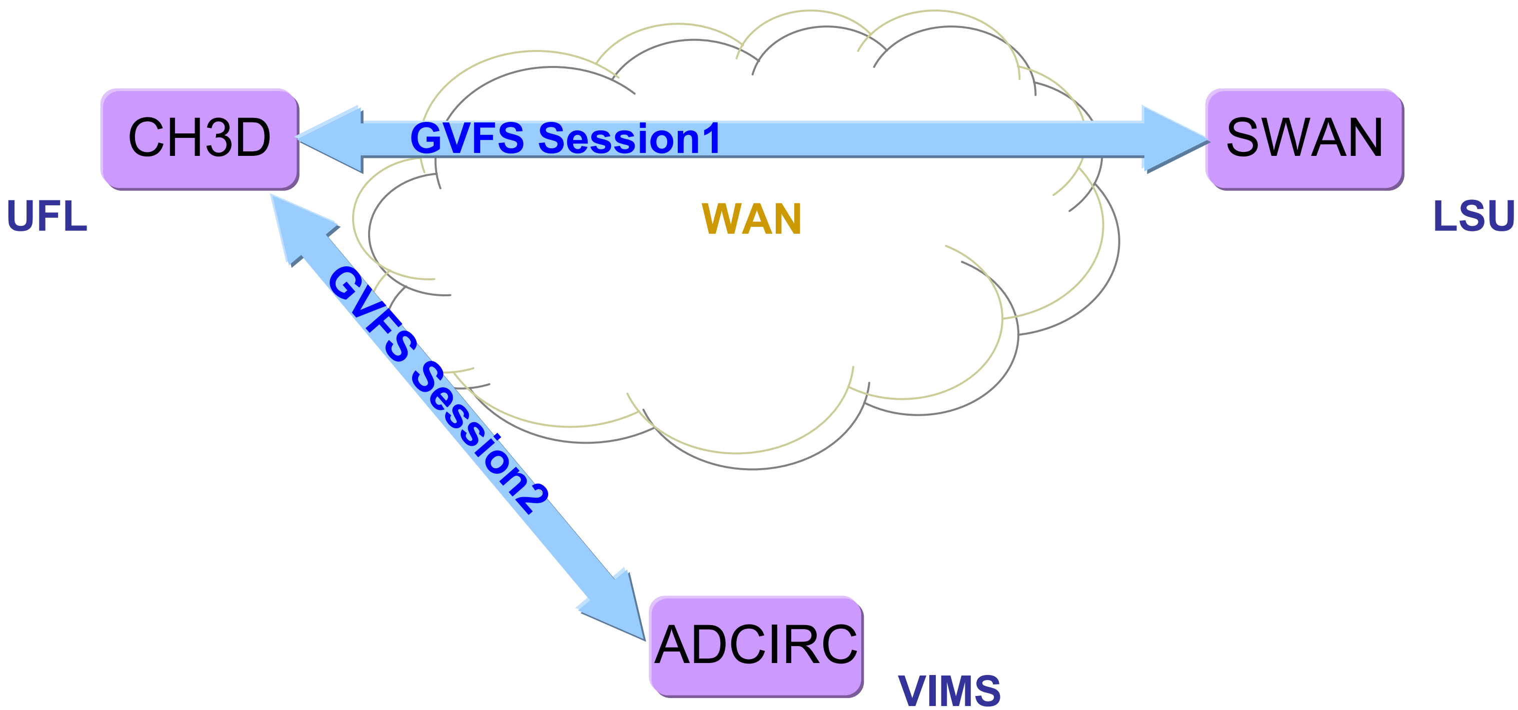
- Shared file system facilitates communication and synchronization between coupled applications
- Distributed file systems in wide-area environments?
  - LAN file systems have shortcomings
  - WAN file systems not widely deployed



Coastal surge coupled modeling

Motivating Example

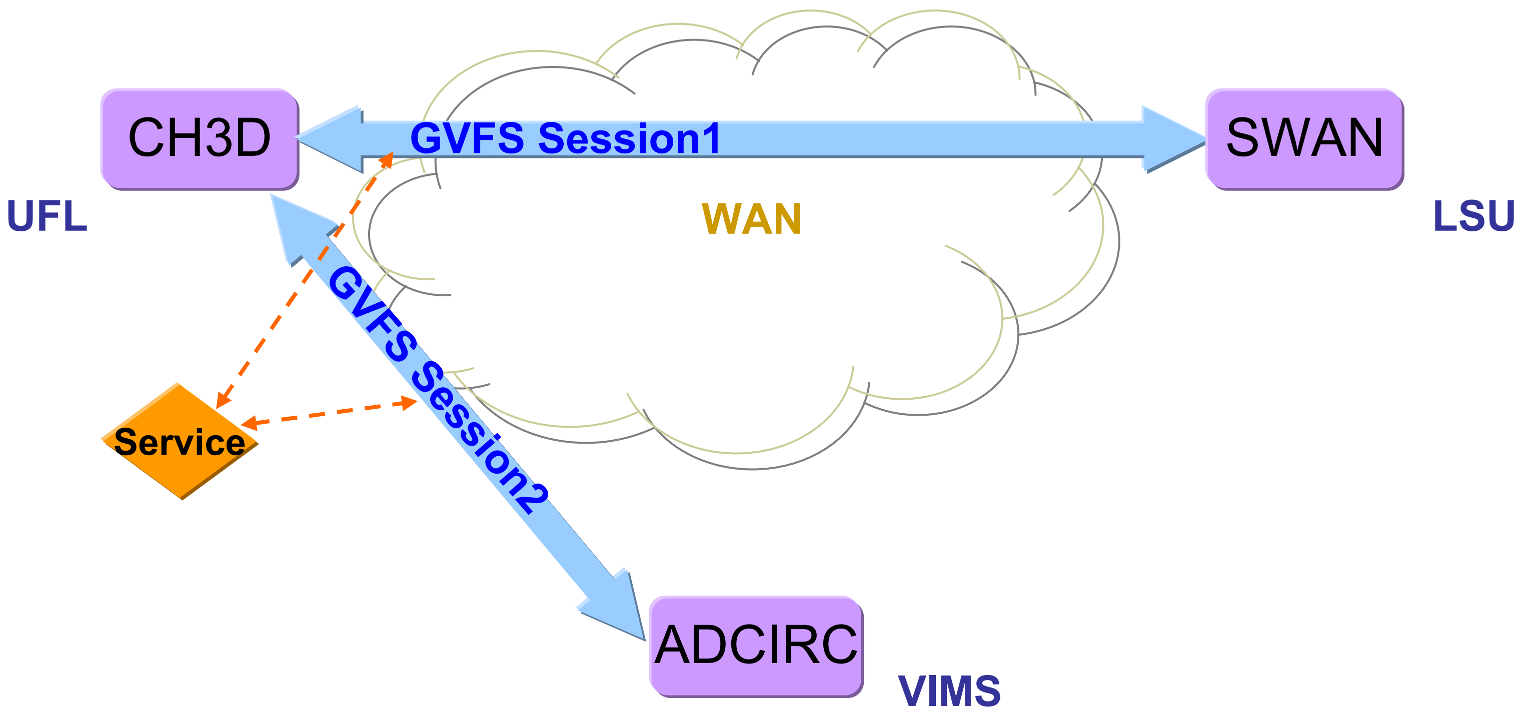
- **G**rid **V**irtual **F**ile **S**ystem (GVFS)
  - Virtualization, user-level proxy, unmodified kernel NFS
  - Cross-domain user identity mapping
  - Performance, security, consistency, reliability enhancements
- Dynamic, independent, application-tailored GVFS sessions



Coastal surge coupled modeling

Motivating Example

- How to manage Grid data sessions
  - Creation, cleanup, isolation, customization …
- WSRF-based data management services
  - Interoperability, flexibility, state management



Coastal surge coupled modeling

Overview

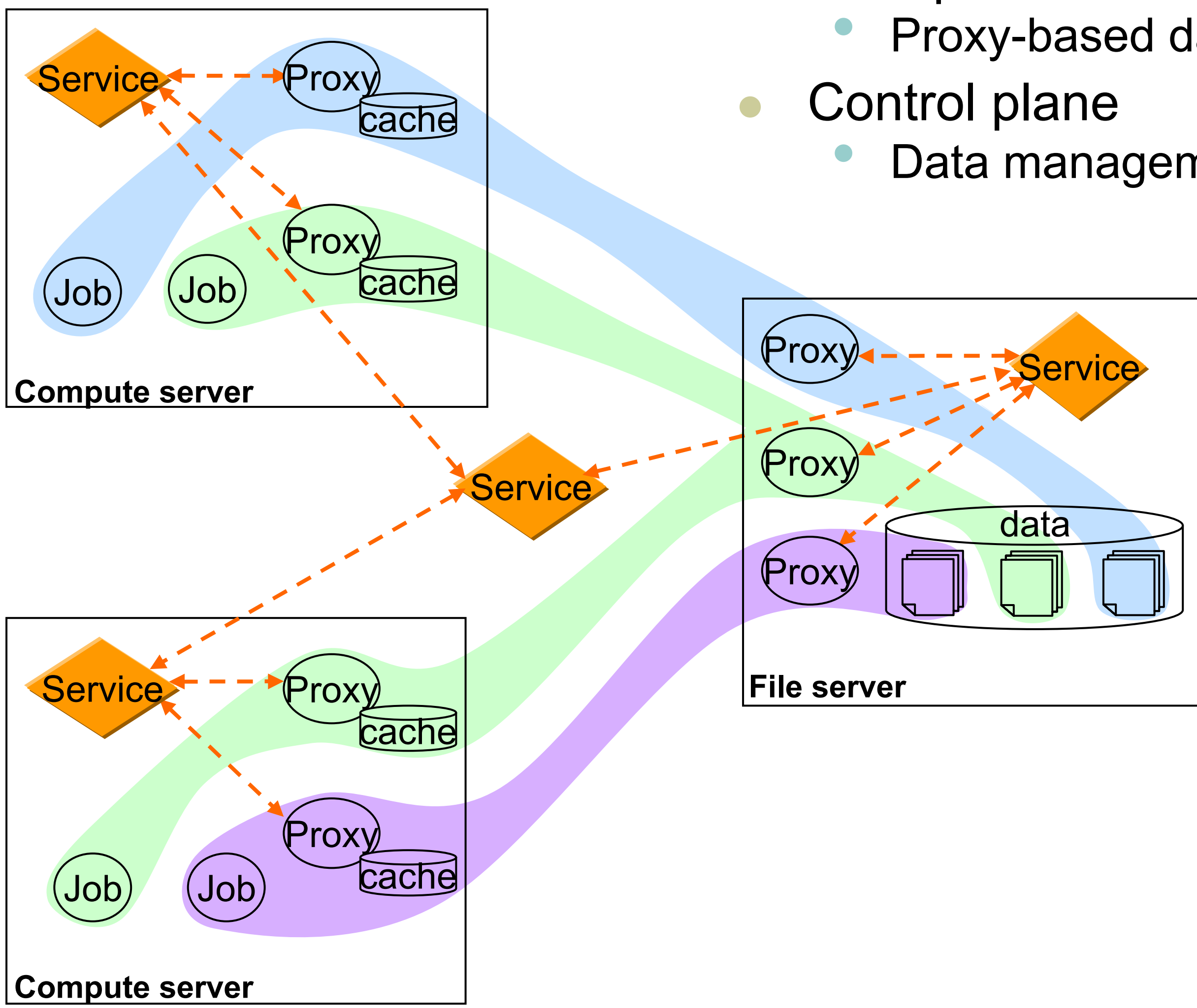
- Goal:
  - Seamless and high-performance data provision for applications in Grid environments
- Challenges:
  - Application transparency for Grid-enabling of a wide range of applications
  - Application-tailored enhancements on performance and reliability for diverse application needs
- Contributions:
  - WSRF-based data management services
  - Enabling of application-tailored grid data sessions

Outline

- Introduction
- Architecture
  - Data Access: Application-Tailored Sessions
  - Control: Data Management Services
- Evaluation
- Summary

Architecture

- Data plane
  - Proxy-based data sessions
- Control plane
  - Data management services

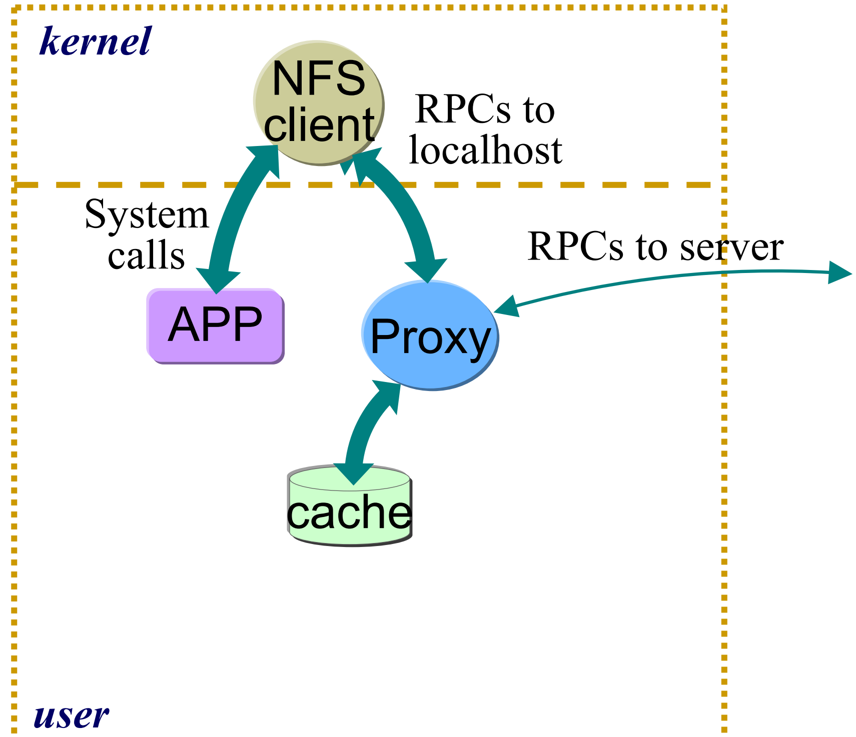


Application-Tailored Data Sessions

- Grid data access
  - Implicit: GVFS proxy RPC interception
    - Partial file transfer, block-based disk caching
    - Configurable cache parameters:
      - Capacity, associativity, read/write, write-through/-back
    - Security mechanisms
      - Session-key authentication, encrypted data channel
  - Explicit: GridFTP/SFTP
    - Full file transfer, file-based disk caching
    - Data accessible through GVFS interface
- Cache consistency models
- Fault tolerance techniques

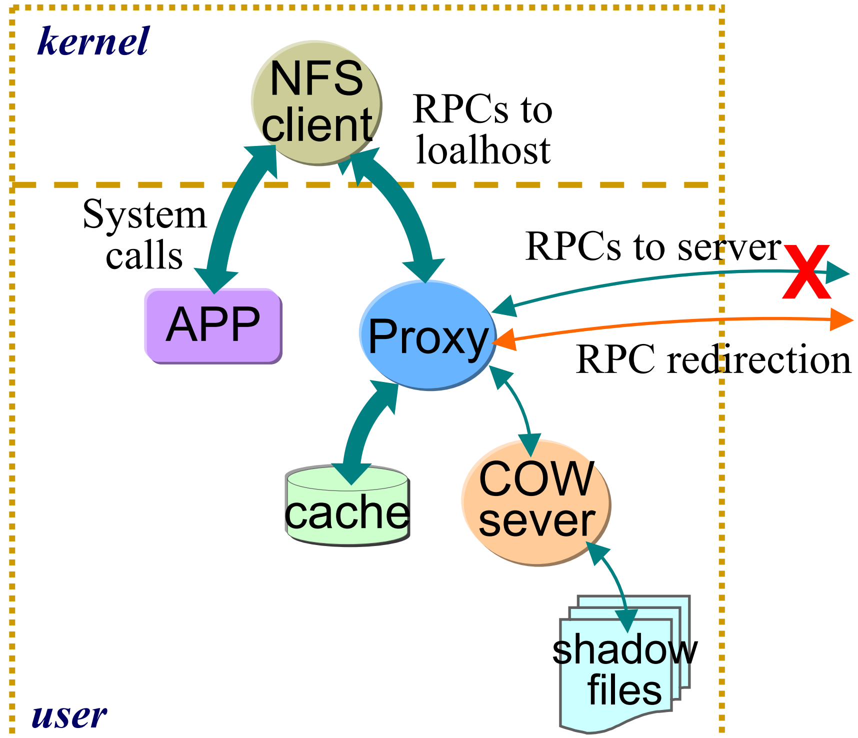
Cache Consistency Models

- Per-session customization
  - Overlaid upon native NFS client polling mechanism
  - Reconfigurable at run-time
- Suitable for various scenarios
  - Single-client sessions:
    - Aggressive read/write caching with write delay
  - Multiple-client sessions:
    - Relaxed polling-based model
    - Strong callback-based model

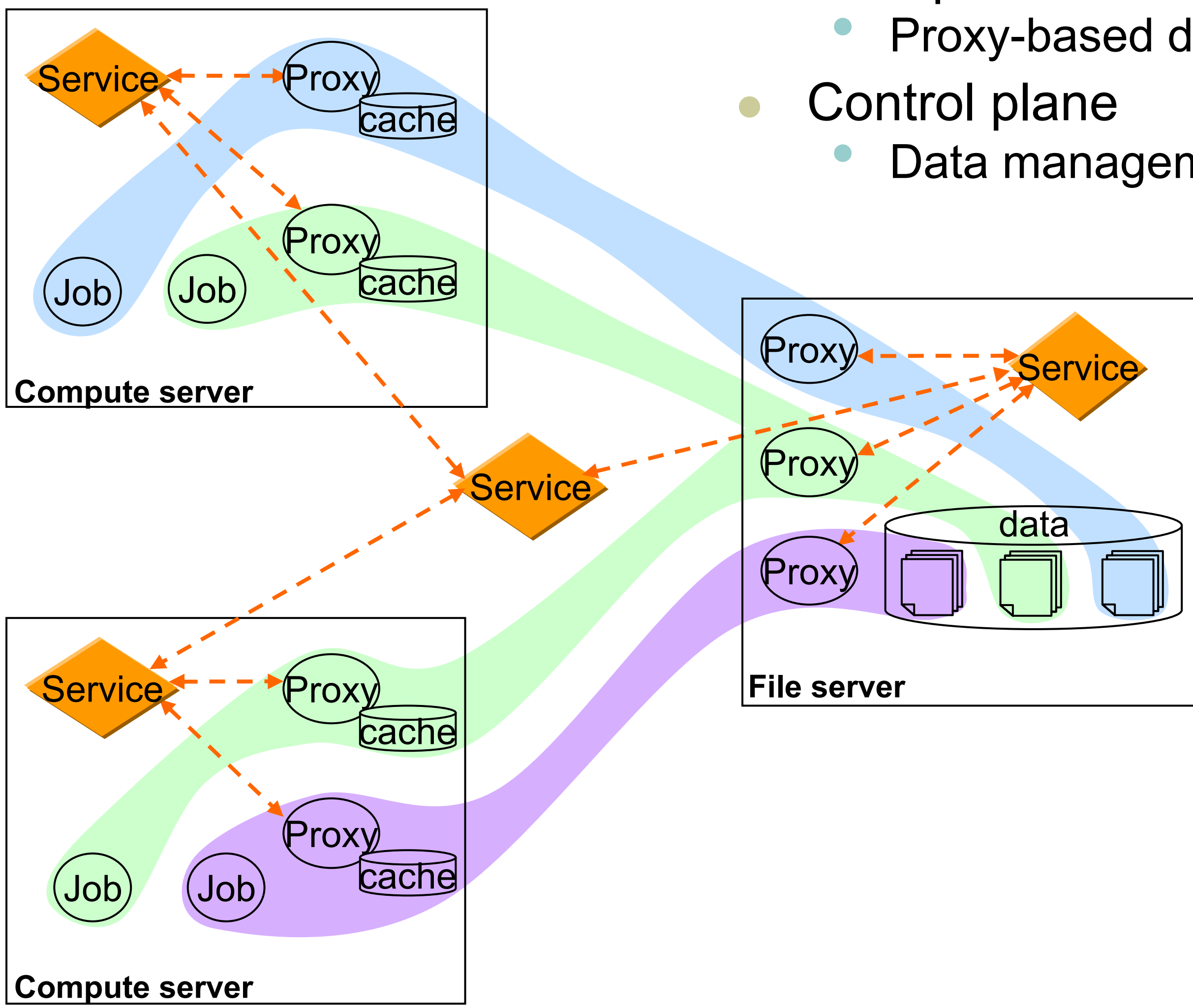


Fault Tolerance

- Copy-on-write file system
  - Fail-over client failures
  - Buffers data modifications on local stable storage
  - Application checkpointed with file system changes consistently
- Session redirection
  - Fail-over server failures
  - Fault detected by RPC timeout
  - Subsequent requests redirected to replica server
  - Proxy remaps file handles transparently from kernel



Architecture



- Data plane
  - Proxy-based data sessions
- Control plane
  - Data management services

Data Management Services

- Service oriented middleware
  - Creation, customization, management of sessions
  - File System Service (FSS)
  - Data Scheduler Service (DSS)
  - Data Replication Service (DRS)
- Built using WS-Resource Framework
  - Interoperability and state management
- Implemented with Perl-based WSRF::Lite
  - WS-Addressing, WS-ResourceProperties, WS-ResourceLifetime, WS-BaseFaults, WS-Security etc.

File System Service (FSS)

- Management of GVFS proxies
- Customization
  - Defined in a configuration file
  - Represented as WS-Resource Property
- Reconfiguration:
  - By signaling proxy to reload configuration file
- Monitoring:
  - By signaling proxy to report accumulated statistics

Configuration File

base_path	/home/cache
session_key	XXYYZZ
acache_enabled	1
dcache_enabled	1
wb_enabled	1
inval_enabled	1
acache_size	65536
acache_asso	8
acache_banks	128
dcache_size	1048576
dcache_asso	16
dcache_banks	512
inval_min	3
inval_max	60

Data Scheduler Service (DSS)

- Manages Grid data sessions
  - Interacting with client- and server-side FSS
  - Session information
    - Represented as WS-Resource Property
    - Stored in MySQL database

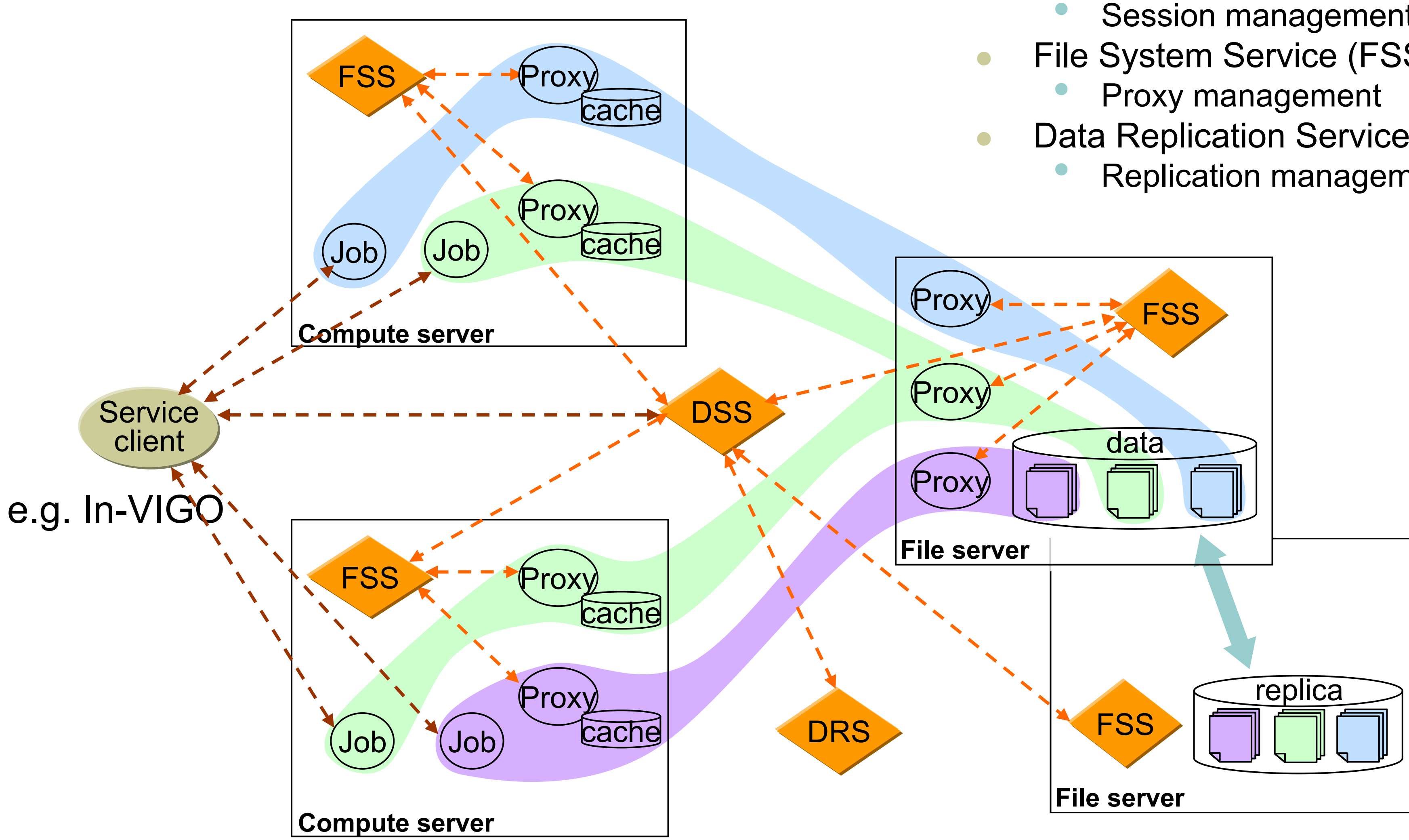
- Resolves conflicts when scheduling a session
  - If another session accesses with write caching
    - Forces it to write back and disable write caching
  - If another session has exclusive access
    - Denies the new session request

Data Replication Service (DRS)

- Manages data replication
  - Replica information represented as WS-Resource Property, stored in MySQL database
- Interacts with DSS for replication and recovery
  - Replication: requests a session for data transfer
  - Recovery: provides replica information to client FSS
- Supports various consistency schemes
  - Uses COW to avoid propagation of writes
  - Active-style or primary-based

Example

- Data Scheduler Service (DSS)
  - Session management
- File System Service (FSS)
  - Proxy management
- Data Replication Service (DRS)
  - Replication management

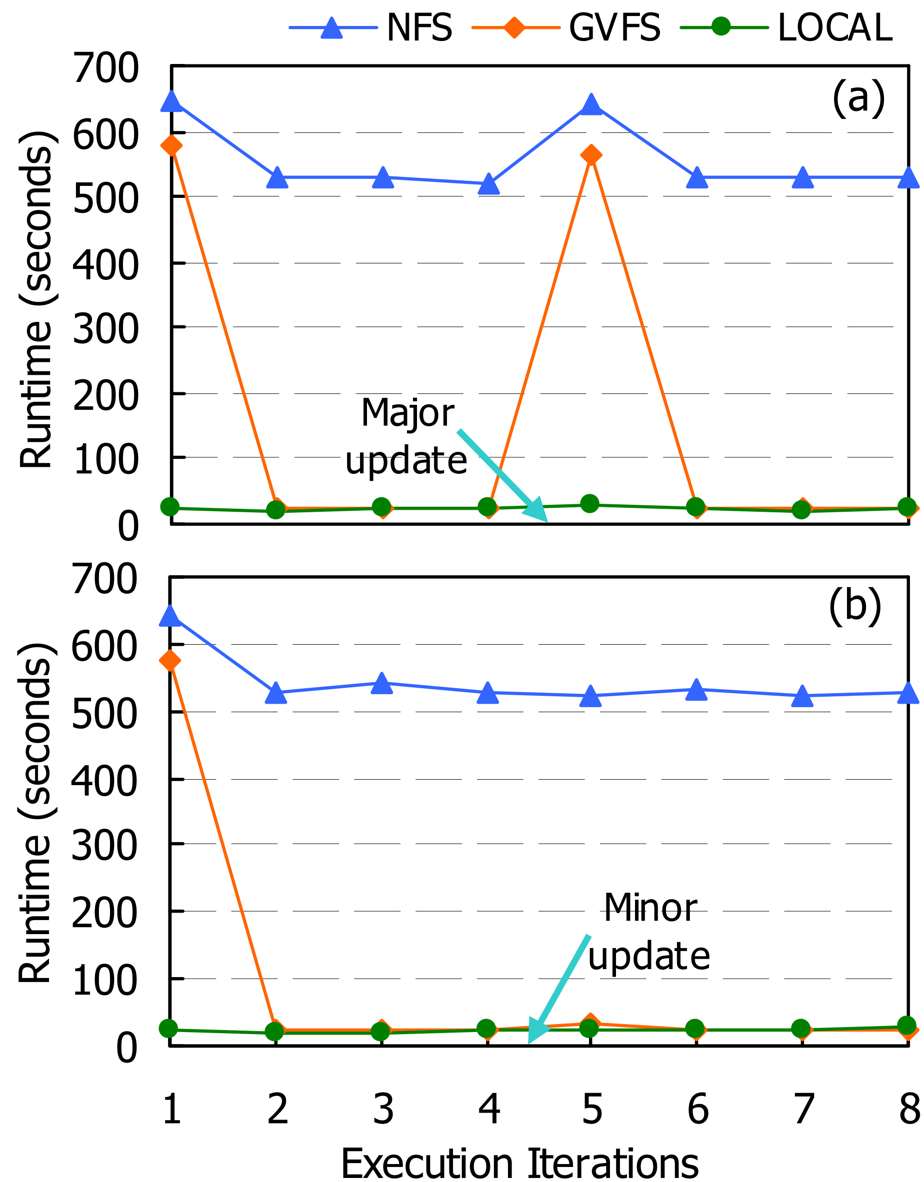


Outline

- Introduction
- Architecture
- Evaluation
- Summary

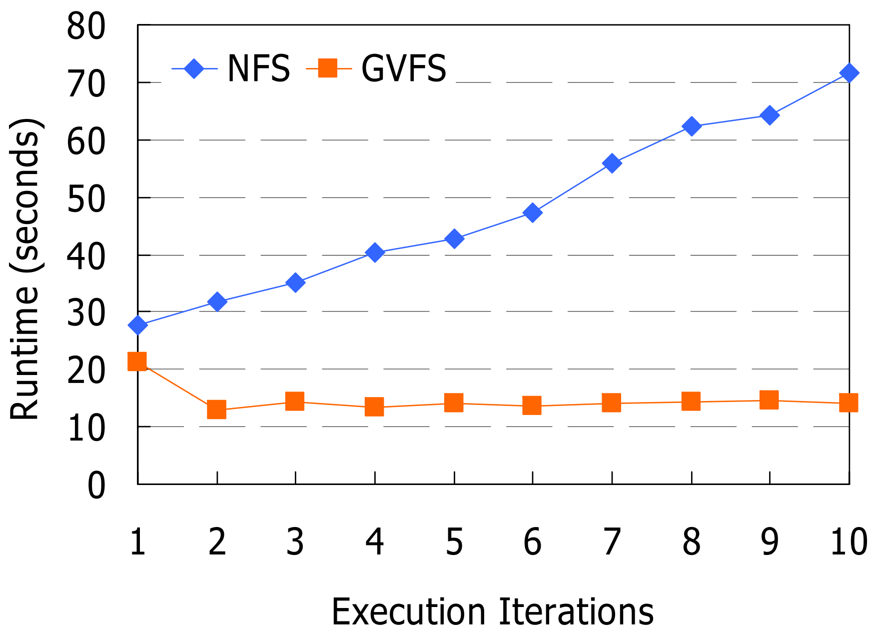
Weak Consistency: Experiment I

- Benchmark:
  - NanoMOS (MATLAB-based 2-D n-MOSFET simulator)
- Scenario:
  - Software accessed by WAN users and updated by local administrator:
    - (a) Major: entire MATLAB
    - (b) Minor: one MATLAB toolbox
  - NFS vs. GVFS
- Observation:
  - With warm disk cache GVFS
    - Filters substantial kernel issued consistency checks
    - Delivers performance close to local disk



Weak Consistency: Experiment II

- Benchmark:
  - CH1D (coupled hydrodynamics simulation and post-processing)
- Scenario:
  - Real-time data accumulated on-site, and processed off-site
    - 30 new inputs available before each run of data processing
  - NFS vs. GVFS
- Observation:
  - As input dataset grows overhead caused by consistency checks:
    - Grows linearly in native NFS,
    - Stays constant in GVFS



Checkpointing and Recovery

- Application:
  - Gaussian (computational chemistry tool)
- Scenario:
  - Client (a virtual machine) is checkpointed, continues to execute and later fails
  - The program changes the state of the file server irreversibly – by deleting temporary files after the checkpointing
- Observation:
  - When the VM is resumed to the checkpoint:
    - Native NFS: stale file handle error; program aborts
    - GVFS with COW: program recovered successfully

Error Detection and Data Redirection

- Application:
  - SPECseis96 (seismic data processing)
- Scenario:
  - File server fails during the program's execution
- Observation:
  - Upon native NFS: program fails (aborts or hangs)
  - Upon GVFS and data replica:
    - Proxy detected the error after a RPC timeout
    - The data request is redirected to the replica within 5 seconds
    - Program continues successfully and is unaware of the failure

Summary

- Problem: Application-transparent and application-tailored Grid data access
- Solution: WSRF-based data management services for application-tailored Grid file system sessions
- Evidence: Experiments based on scientific application execution demonstrate good performance and fault tolerance of GVFS

Related Work

- Grid data management approaches
  - GASS, GridFTP
    - Explicit transfer via middleware or use of specialized API
  - Condor, BAD-FS
    - On-demand remote data access by interception of system call
    - Control caching, consistency and fault tolerance to middleware
  - LegionFS, Avaki's Data Grid Access Servers
    - Access of Grid data based on NFS
- WSRF-based Grid middleware
  - Globus Toolkit 4 based data management middleware
  - WSRF.NET based (remote job execution grid)
  - WSRF::Lite based (WEDS)

Acknowledgments

- In-VIGO team
  - http://invigo.acis.ufl.edu
- Dr. Peter Dinda
- Dr. Peter Sheng, SCOOP resources

- NSF Middleware Initiative
- NSF Research Resources
- IBM Shared University Research
- VMware

- **Questions?**

References

[FGCS'05] S. Adabala, V. Chadha, P. Chawla, R. Figueiredo, J. Fortes, I. Krsul, A. Matsunaga, M. Tsugawa, J. Zhang, M. Zhao, L. Zhu, and X. Zhu, "From Virtualized Resources to Virtual Computing Grids: The In-VIGO System", special issue on Complex Problem-Solving Environments for Grid Computing, Vol 21/6, 2005.

[CC'04] R. Figueiredo, N. Kapadia, J. Fortes, "Seamless Access to Decentralized Storage Services in Computational Grids via a Virtual File System", In Cluster Computing, 2004.

[HPDC'04] M. Zhao, R. Figueiredo, "Distributed File System Support for Virtual Machines in Grid Computing", In Proceedings of 13th IEEE International Symposium on High Performance Distributed Computing, June 2004.

WSRF::Lite: An Implementation of the Web Services Resource Framework
http://www.sve.man.ac.uk/Research/AtoZ/ILCT

In-VIGO:

In-VIGO prototype can be accessed from http://invigo.acis.ufl.edu; courtesy accounts available.

Future Work

- Extensive evaluation and performance tuning of service-based middleware
- Use of application profiling to assist the customization of Grid data sessions
- Fine grained replication management and load balancing schemes