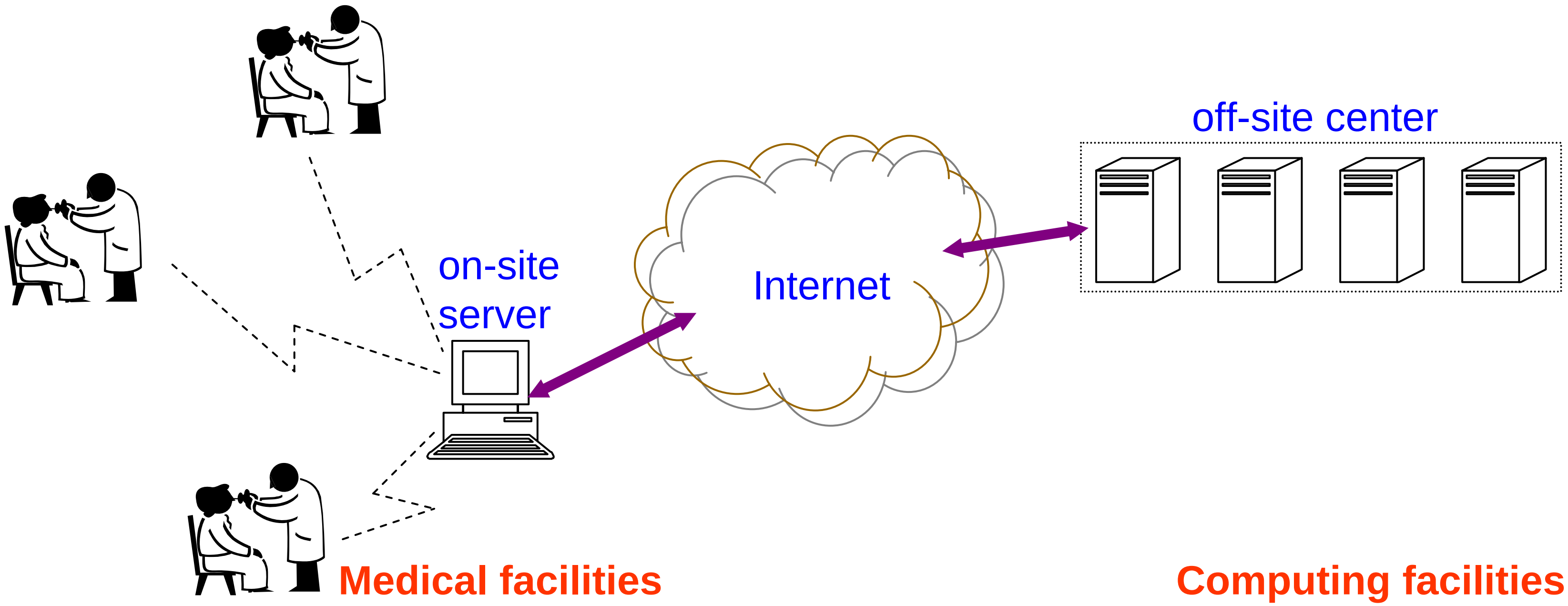
Support for Data-Intensive, Variable-Granularity Grid Applications via Distributed File System Virtualization: A Case Study of Light Scattering Spectroscopy

Jithendar Paladugula, Ming Zhao, Renato Figueiredo

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

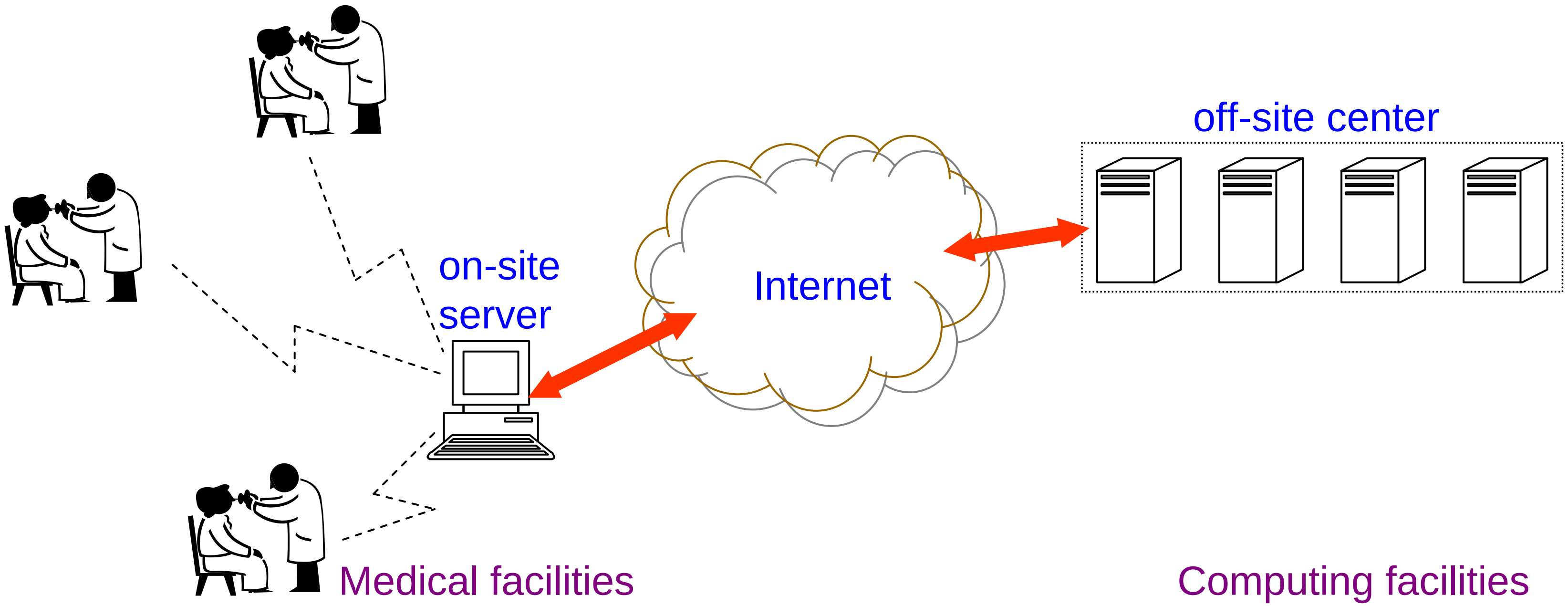
Overview

- **Goal**: Support for large-scale, distributed biomedical applications on computational Grids
  - Network/Grid computing model
  - Data access at variable granularities



Overview

- **Challenge**: High performance and seamless data management
- **Contribution**: The integration of Light Scattering Spectroscopy (LSS) analysis with Grid Virtual File System (GVFS)

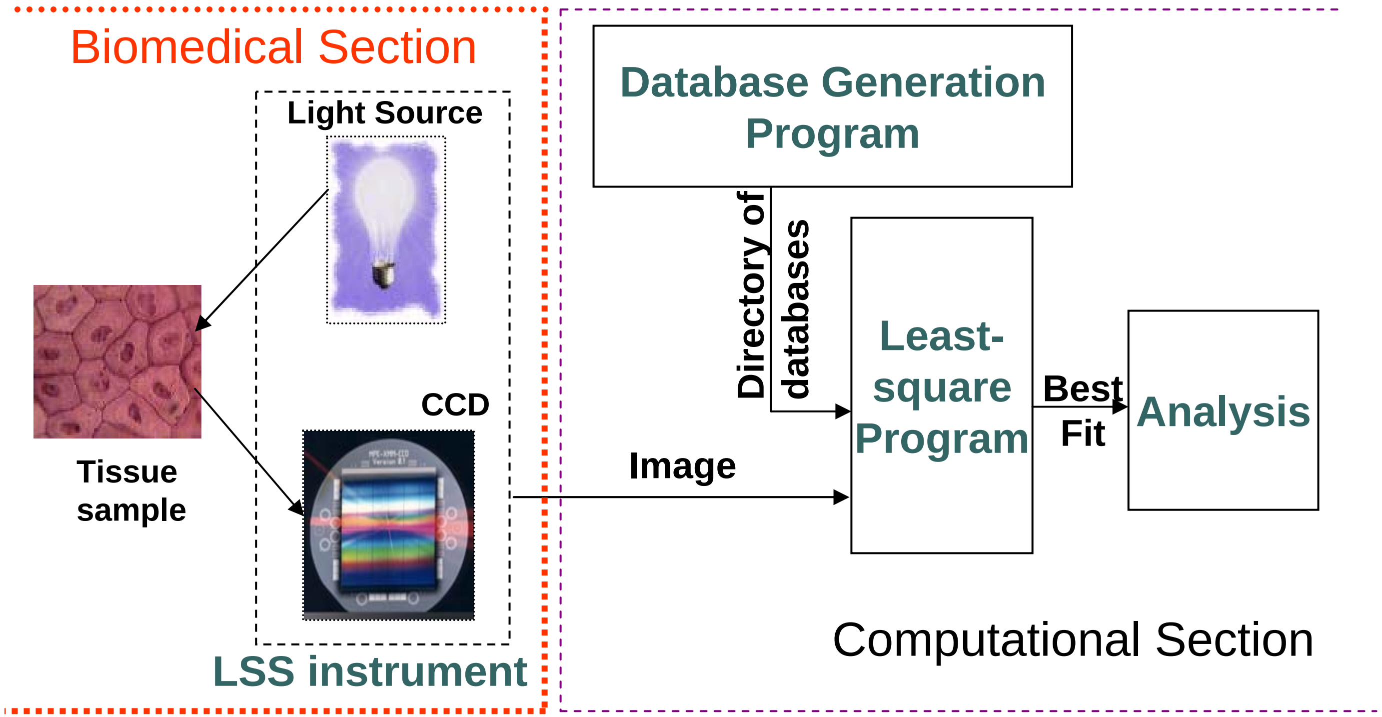


Outline

- Background
- Implementation
- Evaluations
- Summary

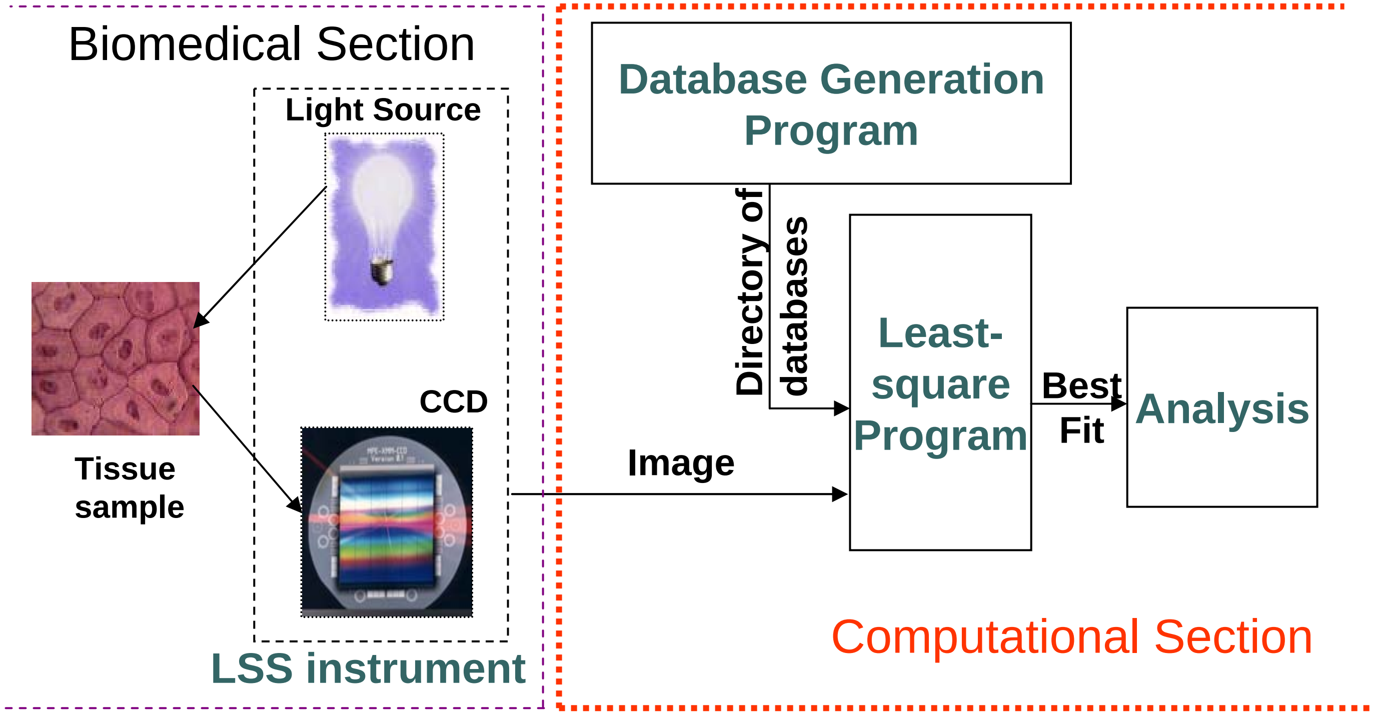
Light Scattering Spectroscopy (LSS) [1][2]

- Probes the structure of living cells without tissue removal
- Helps in non-invasive detection of precancerous changes in human epithelium



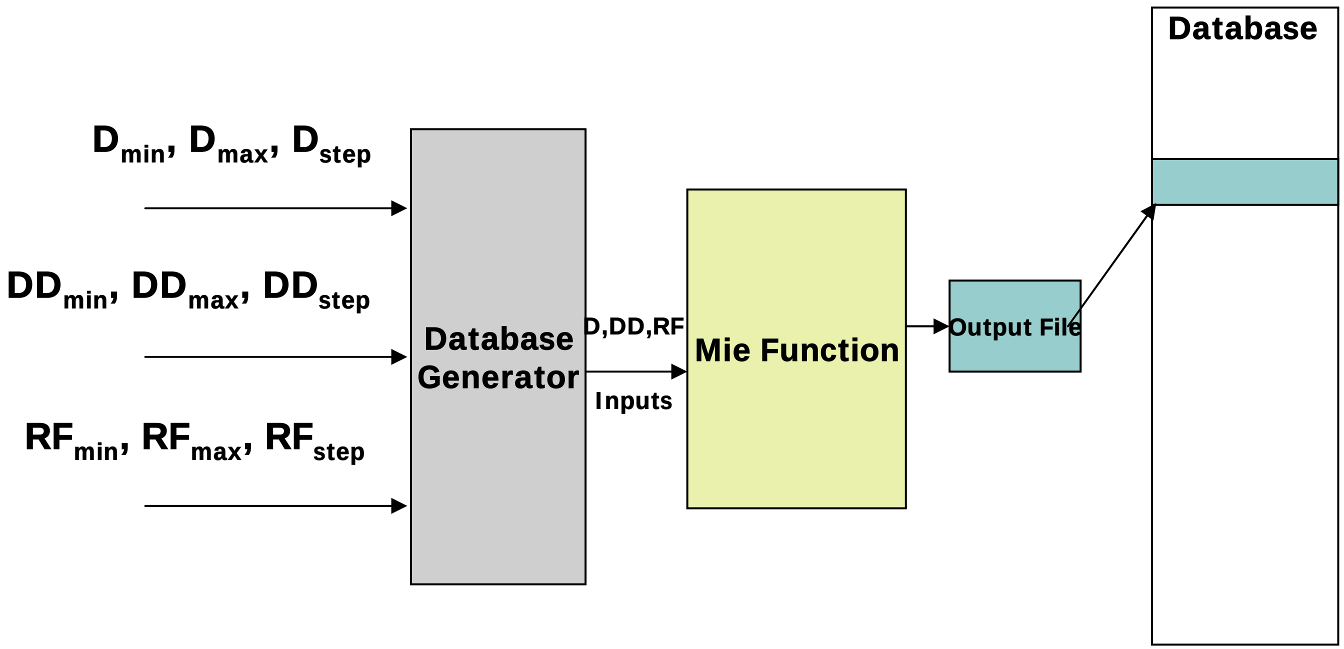
LSS Analysis

- Obtains parameters (size and refractive index) from spectrum
- Approximated using lookup on Mie-theory spectra database
- High accuracy => large database, intensive computation

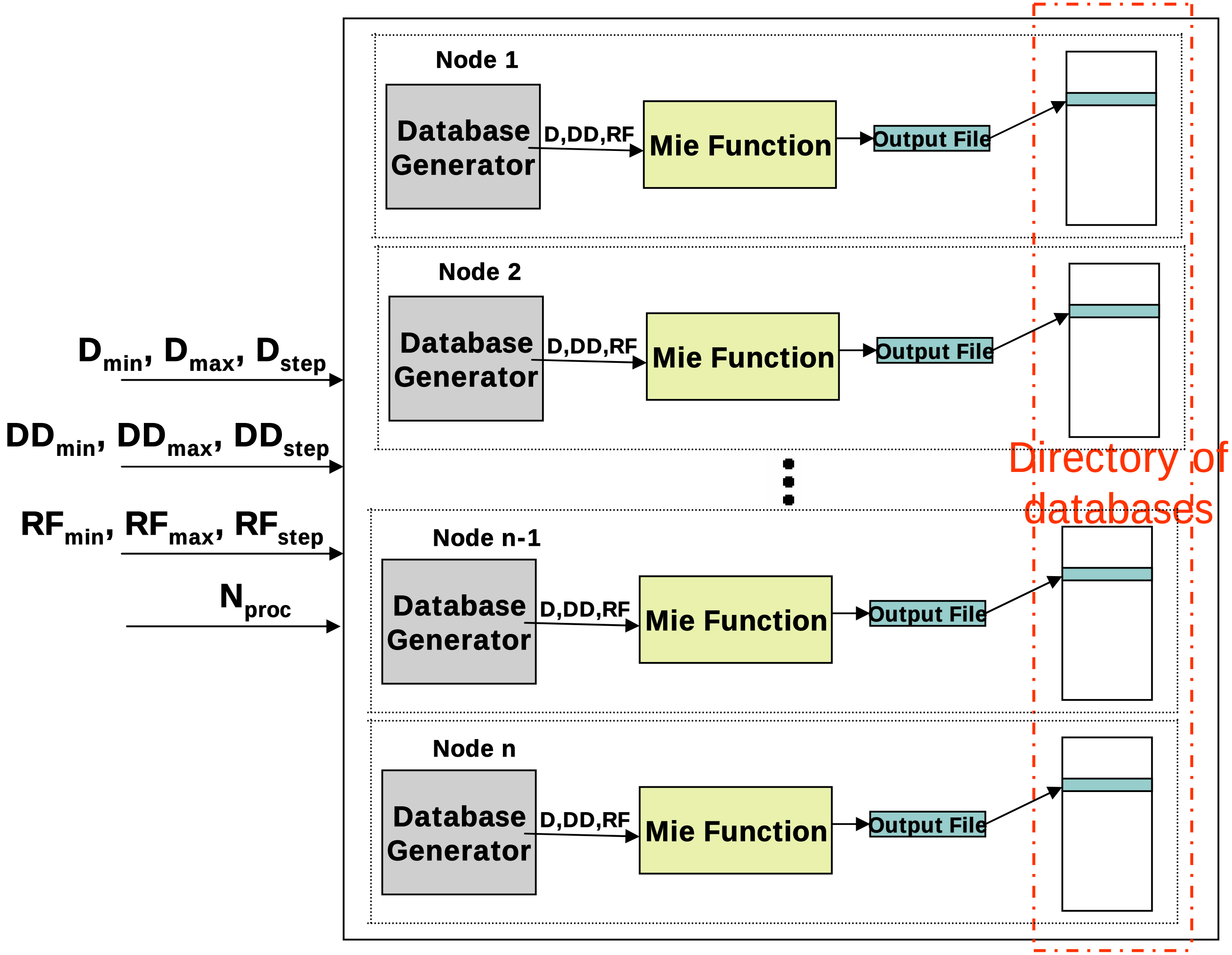


Database Generation

- Databases of LSS spectra over a range of diameters, diameter deviations, refractive indices are generated.
- The Mie function output file is normalized and appended as a record to the database.

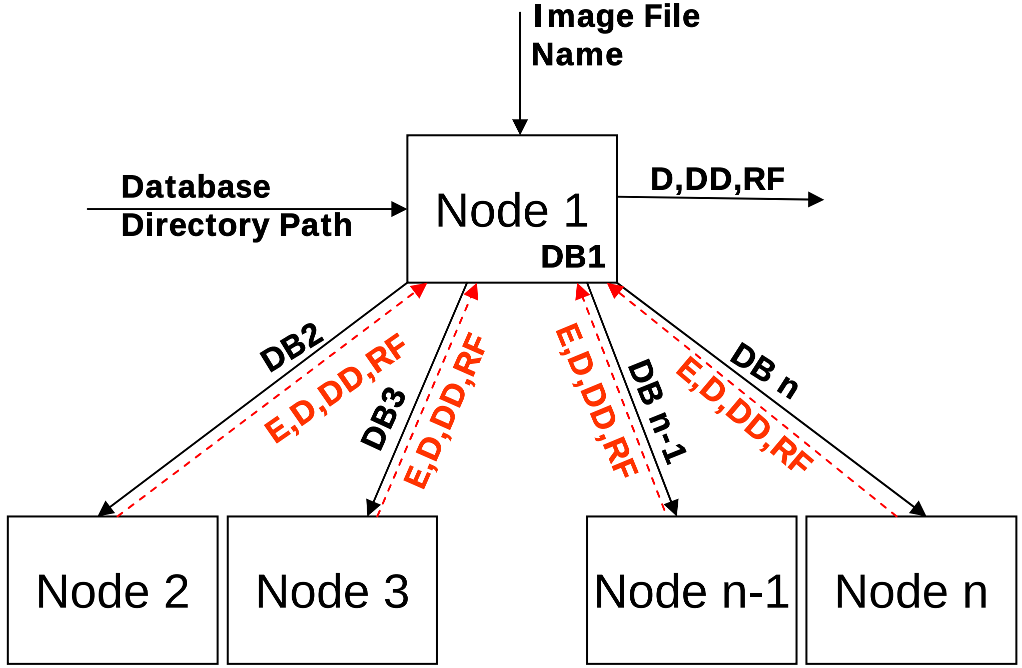


Database Generation in Parallel



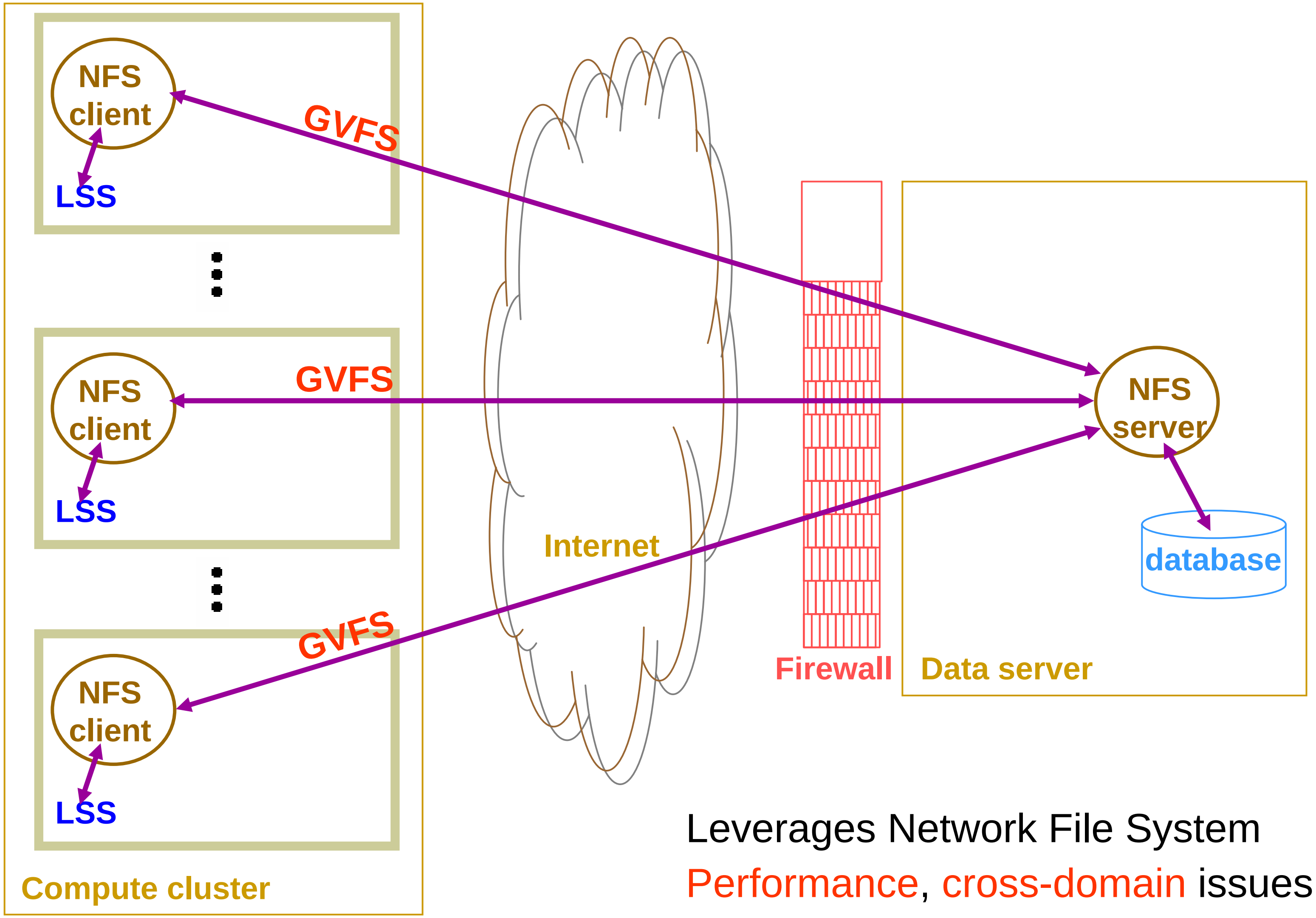
LSS Analysis in Parallel

- Parallelized across database records
  - Large database size
  - Fit into cache
- MPI for coordination
  - Master-slave strategy
- File system I/O for access of databases
  - Simplifies programming
  - Exploits GVFS support



D – Diameter, DD – Diameter Deviation,
RF – Refractive Index, E – Error, DB - Database

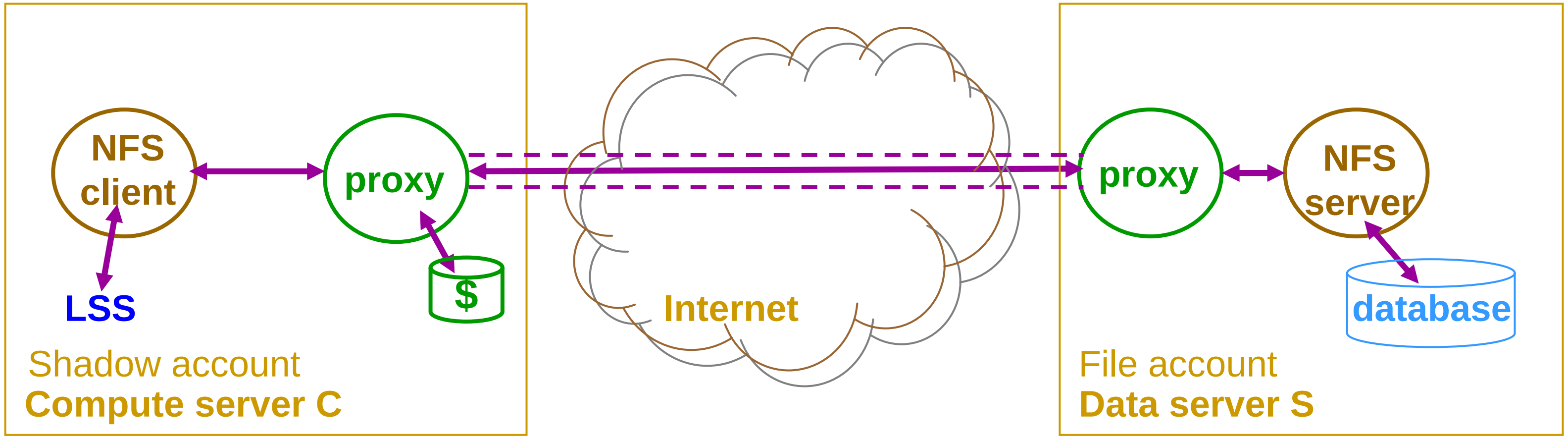
Integration with Grids



Leverages Network File System

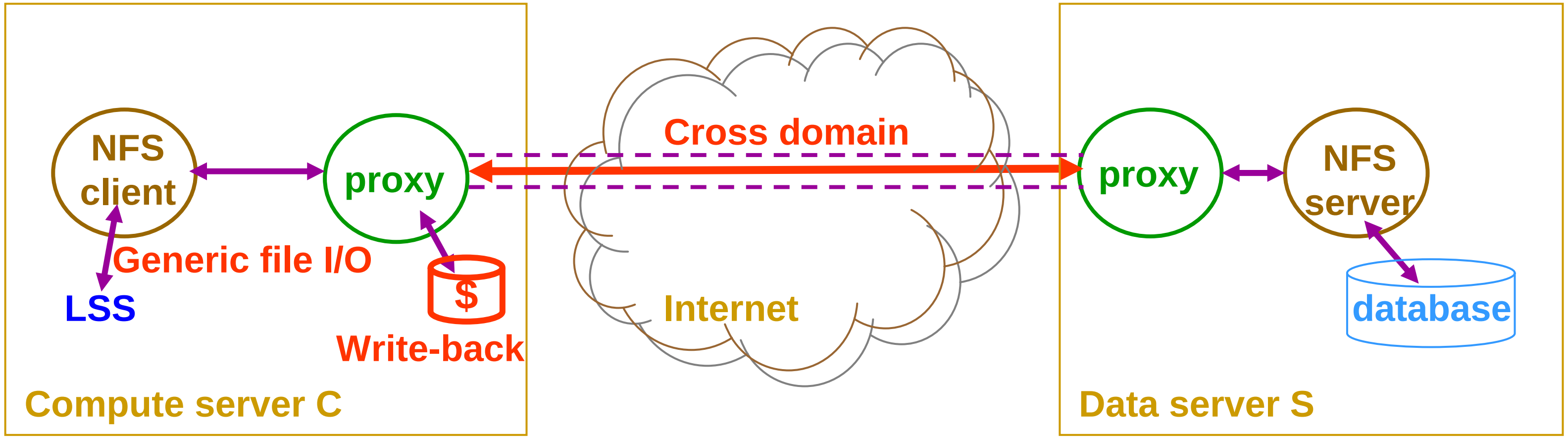
Performance, cross-domain issues

Grid Virtual File System (GVFS)



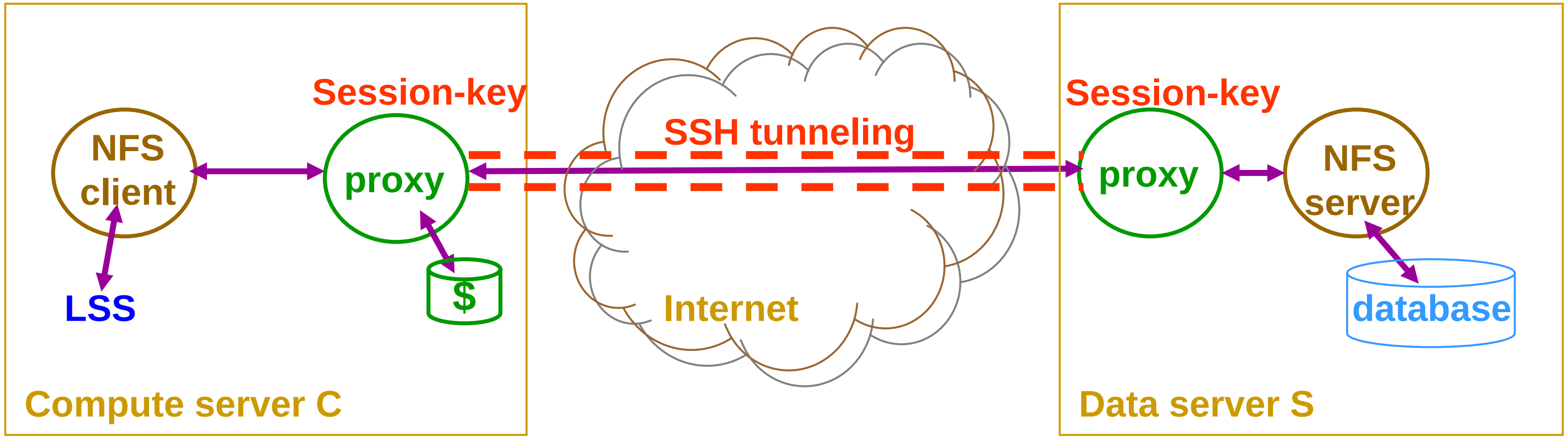
- Logical user accounts [3] and Virtual file system [4]
  - NFS call forwarding via middle tier user-level proxy
  - On-demand, partial, user-transparent data transfer
- Performance: client-side proxy disk caching
- Security: SSH tunneling of RPC connections and cross-domain session-key based authentication

GVFS Support for LSS



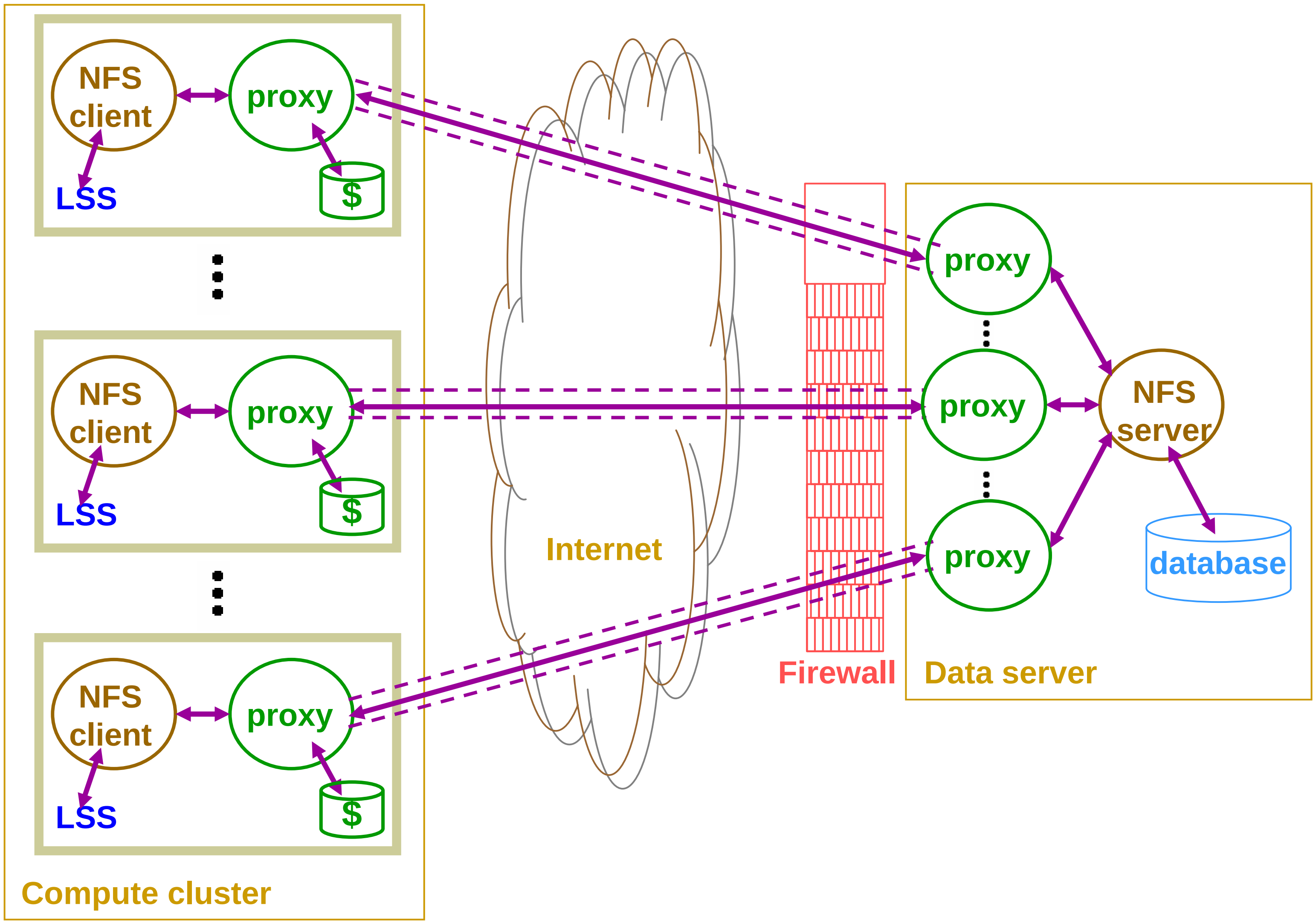
- File I/O across wide area environment
  - Simplifies programming, reduces communications
  - User transparent, cross-domain data access
- Network latency hiding by disk caching
  - Exploits temporal locality of databases across LSS runs
  - Employs write-back to hide write latency and avoids transfer of temporary data

GVFS Support for LSS

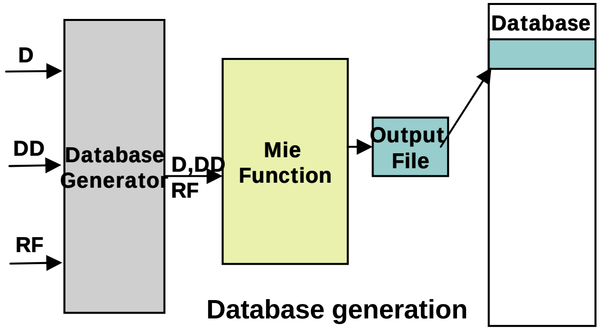


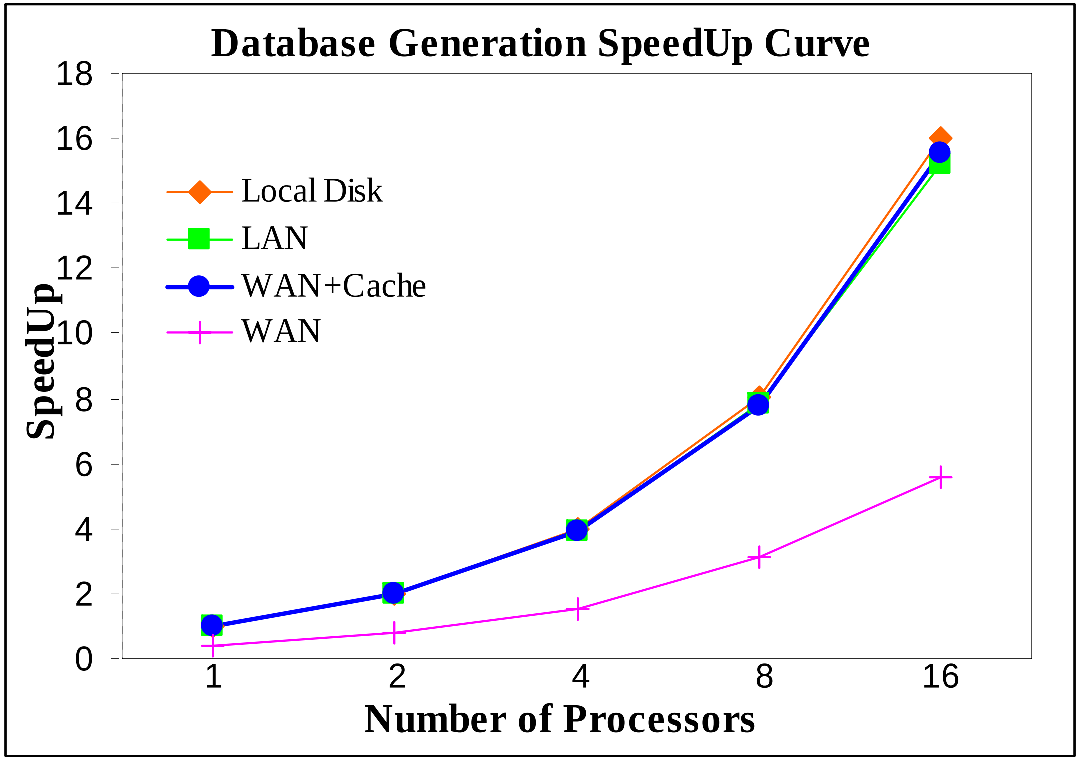
- On-demand data access at variable granularity
  - Fast response: sampling down databases
  - High accuracy: large databases
- Private data access via encrypted data channels
  - SSH tunneling
  - Inter-proxy session-key authentication

Integration with Grids



Database Generation Results

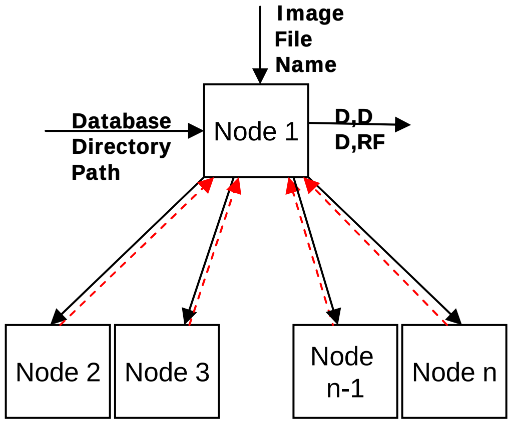




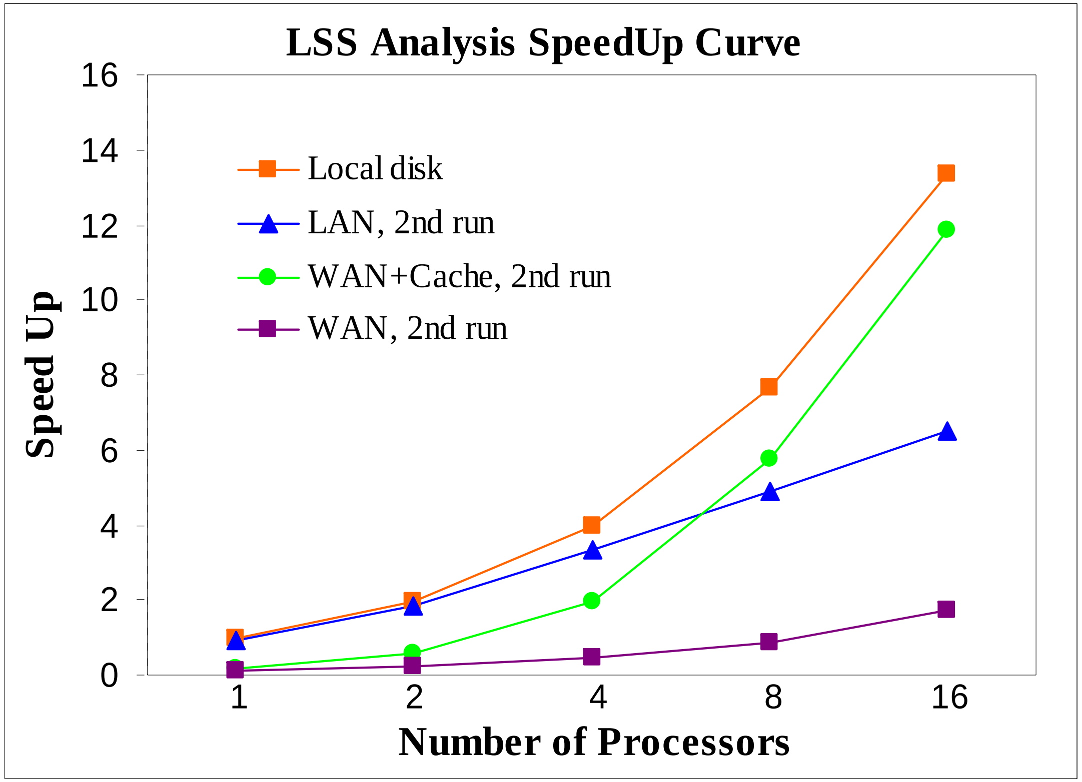
Speedup plot for parallel database generation

- Databases are stored in Local disk, LAN and WAN data servers
- Proxy disk cache is disabled (WAN), or enabled with write-back policy (WAN+Cache)

LSS Analysis Results



LSS analysis



Speedup plot for parallel LSS analysis

- Databases are stored in Local disk, LAN and WAN data servers
- Proxy disk cache is disabled (WAN) or enabled (WAN+Cache)

Variable Granularity

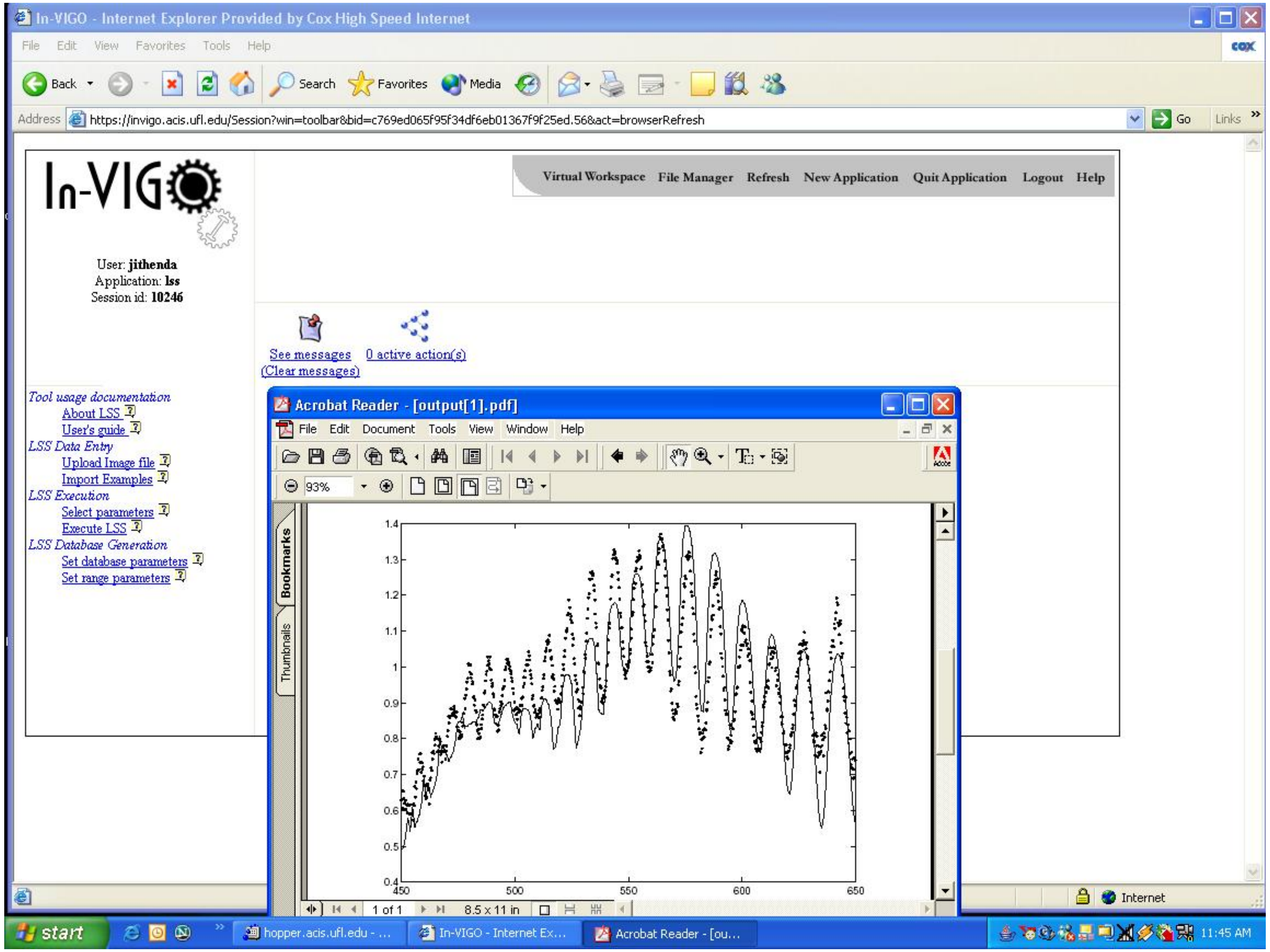
Least-square error, WAN execution time and number of NFS data blocks transfers for database sampling with 16 nodes

Sampling Interval	LSS Error	Time (seconds)	Number of Blocks
1	2.899	793	14666
5	2.9	700	14662
10	2.902	432	6894
20	2.916	323	3622
40	2.934	152	1856

- Low accuracy analysis by sampling down the databases
- A sampling interval of “n” indicates that “n” records are skipped before reading another record in the database

Integration with In-VIGO

- Facilities:
  - Upload files
  - Import examples
  - Generate database
  - Execute LSS
  - PDF outputs
  - File manager



In-VIGO [5]: Virtualization middleware for computational Grids

Related Work

- GEMSS (Grid Enabled Medical Simulation Services) [6]
  - Grid middleware which provides grid services for medical applications
  - Mainly focuses on the computational services for the applications

- ARAMIS (A remote Access Medical Imaging System) [7]
  - Provides an object-based user interface
  - ARAMIS propose two levels of network:
    - High-speed, fast-access network to support transport of large volumes of data (between databases and servers)
    - Low bandwidth network for transport between the servers to user's workstation

Conclusions

- A case study for integration of biomedical applications with Grid environments
  - Light Scattering Spectroscopy application deployed in network/Grid computing model
  - Computational/synchronization requirements addressed using MPI
  - Communication requirements are met by the use of Grid Virtual File System
    - Variable granularity
    - Performance

Ongoing and Future Work

- Collaboration with Northwestern University Biomedical Engineering
- Experiments with actual tissue data
- Interface improvements based on user feedback
- Integration with data collection at LSS instruments

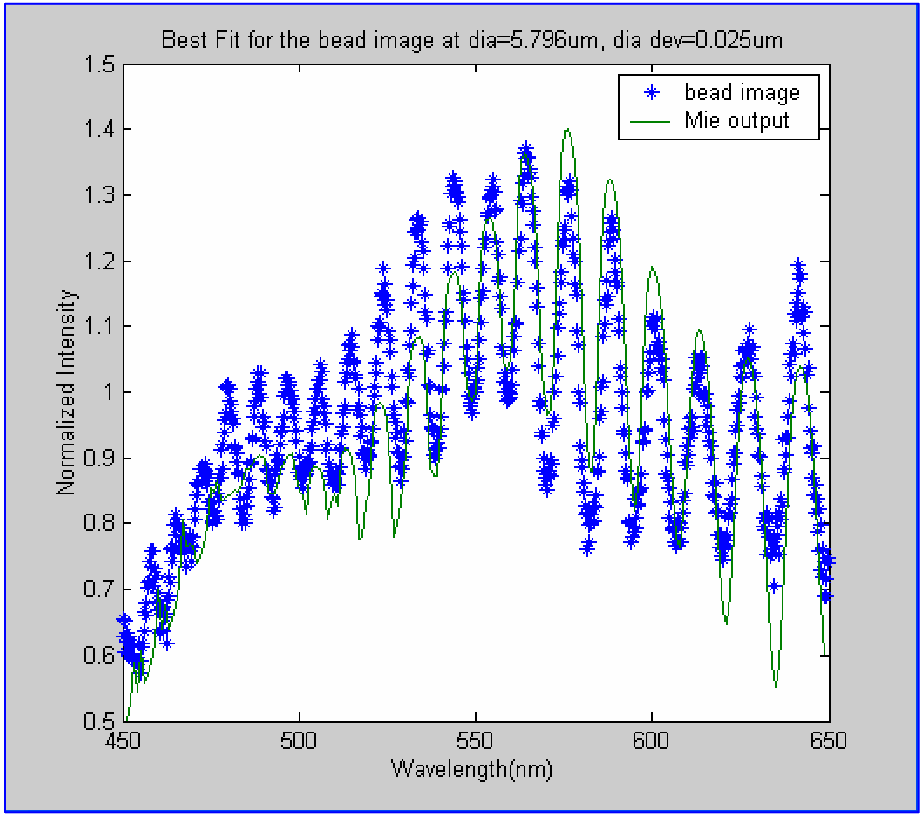
References

1. Backman V, et al. “Detection of preinvasive cancer cells in situ”, *Nature*, 406, 35-36 (2000).
2. Backman V, et al, “Imaging human epithelial properties with polarized light-scattering spectroscopy”, *Nature Medicine*, 7, 1245-1248 (2001).
3. N. Kapadia, R. Figueiredo and J. A. B. Fortes, “Enhancing the Scalability and Usability of Computational Grids via Logical User Accounts and Virtual File Systems”, *Proceedings of HCW at IPDPS*, April 2001.
4. R. Figueiredo, N. Kapadia and J. A. B. Fortes. "The PUNCH Virtual File System: Seamless Access to Decentralized Storage Services in a Computational Grid", *Proceedings of HPDC*, August 2001.
5. 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”, to appear, *Future Generation Computing Systems* (in press), 04/2004 .
6. Berti, G., Benker et al, “Medical Simulation Services via the Grid”, in *Proceedings of HealthGrid Workshop 2003*, Lyon, France, 2003.
7. D. Sarrut and S. Miguet, “ARAMIS: A Remote Access Medical Imaging System”, *International Symposium on Computing in Object-Oriented Parallel Environments*, San Francisco, USA, Dec. 1999.

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

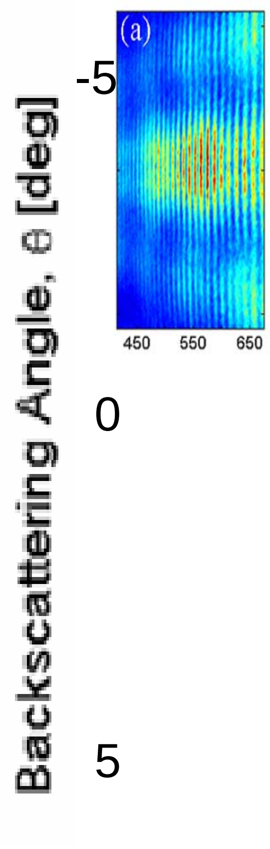
Experimental Evaluation

- Image
  - Polystyrene beads suspended in water
    - Diameter= 5.8um
    - Diameter Deviation= 0.02um
- Generated databases
  - Diameter : 5.65um to 5.97um in 0.0005um steps
  - Diameter Deviation : 0.005um to 2.5um in steps of 0.005um
  - Refractive Index (0)
- LSS analysis best fit
  - Diameter= 5.796um
  - Diameter deviation= 0.025um



LSS Image

413.486	1	1.013e-002
413.743	1	1.127e-002
413.999	1	1.626e-002
414.256	1	3.318e-002
414.512	1	1.571e-002
414.769	1	1.611e-002
415.025	1	2.057e-002
415.282	1	1.622e-002
415.538	1	5.510e-003
415.795	1	1.058e-002
416.052	1	1.501e-002



Database Format

100	#Represents the number of records in the database
400 700 1	#Represents the minimum, maximum and step wavelengths
-5 5	#Represents the minimum and maximum scattering angle
0	#Represents the azimuth angle
1	#Represents the distribution
1.334	#Represents the refractive index of the medium
2	#Represents the width of data points
5.6 0.02 1.1	#Represents the diameter, diameter deviation and refractive index of the first record
400 0.57	#Represents the various data points for the record
401 0.67	

---	
550 1.00	

---	
699 1.3	
700 0.8	

Computational Requirements

- Storage requirements
  - Database requires TBytes of storage
    - Diameter : 0.1um to 20um in steps of 0.005um
    - Diameter Deviation :0.1um to 5um in 0.005um steps
    - Refractive Index : 1.02 to 1.1 in steps of 0.0005
- Processing requirements
  - High accuracy analysis requires Peta-order number of operations
- Solution : Parallel computing on workstation clusters using MPI

Execution Times (seconds) for LSS Analysis

#Proc	Local Disk	LAN		WAN		WAN +C		
							2nd run	
		1st run	2nd run	1st run	2nd run	1st run	mount	un-mount
1	1318	1404	1396	13473	11860	12465	7001	7369
2	664	735	718	5961	5883	5979	2204	2225
4	333	432	397	2992	2986	3044	674	1496
8	172	301	269	1993	1482	1580	228	317
16	99	234	203	817	755	804	111	183

Execution Times (seconds) for Database Generation

#Proc	Local Disk	LAN		WAN		WAN +C (WT)		WAN + C (WB)
		1st run	2nd run	1st run	2nd run	1st run	2nd run	1st run
1	8839	8988	8977	22914	22765	22935	22752	9016
2	4417	4491	4488	11693	11550	13002	11776	4493
4	2212	2253	2251	5910	5790	5971	5775	2249
8	1104	1134	1132	2954	2849	2894	2854	1137
16	553	583	576	1685	1595	1503	1445	570