High-Bandwidth Remote Parallel I/O with the Distributed Memory Filesystem MEMFS

Jan Seidel Rudolf Berrendorf
Marcel Birkner Marc-André Hermanns

September 20th, 2006

Fachhochschule
Bonn-Rhein-Sieg

Forschungszentrum Jülich
in der Helmholtz-Gemeinschaft

VIOLA

Outline

Introduction

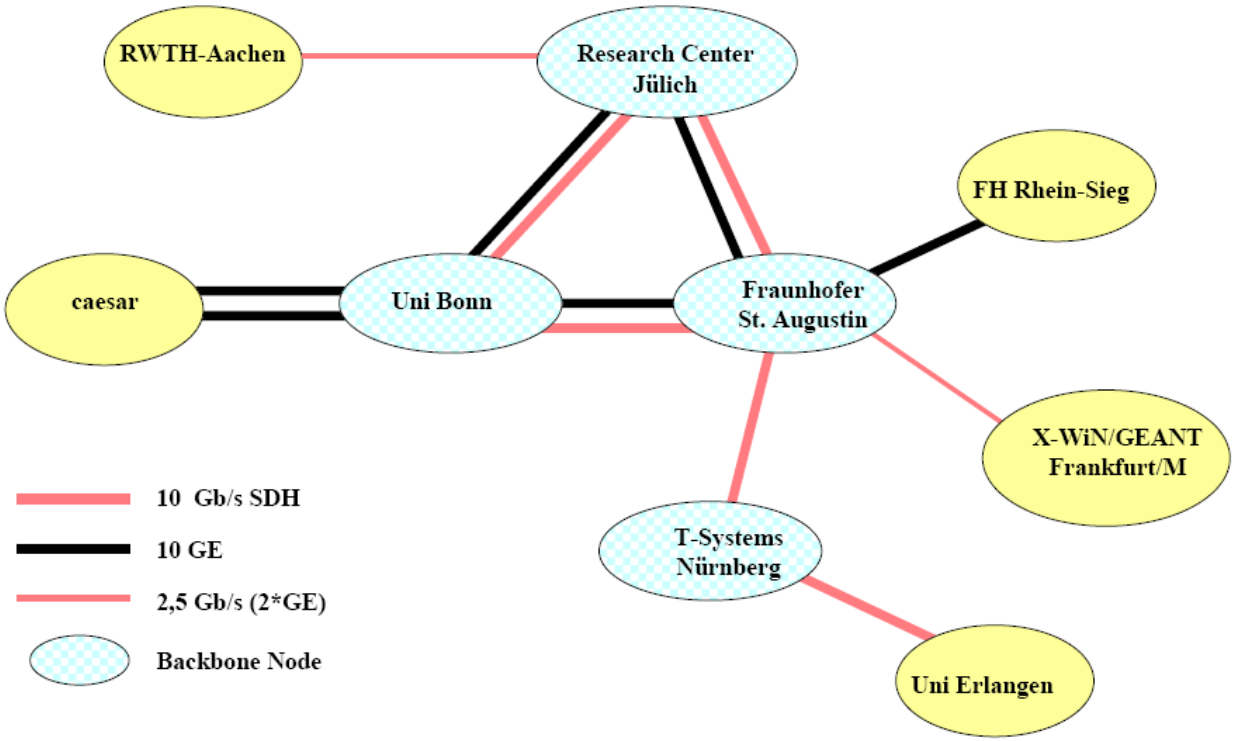
The I/O Bottleneck

- Big advance in computational power of parallel systems
- "Bigger" problems can be solved
- More data is being processed, needs to be transferred from / to storage devices
- Burst transfer of large amounts of data
  - Limited by the bandwidth of common disk-based I/O devices used in many clusters
    - Can not keep up with evolution of computational power

→ Access to the data can become the bottleneck of scientific applications

The VIOLA Project

- German network testbed for grid applications
- Several clusters connected by dedicated 10 GBit/s optical WAN connections to form a grid
- Up to 100 kilometres distance between clusters



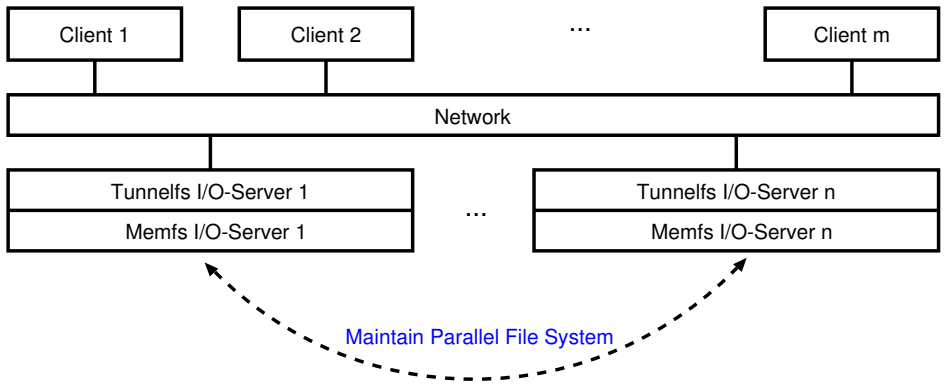
The VIOLA Parallel I/O Subproject

- Subproject develops middleware to comfortably run parallel applications on multiple clusters
- Provide high-bandwidth transparent access to storage of remote hosts using MPI-IO
- Designed for usage in reconfigurable grid environments on a demand basis
- MP-MPICH used for inter-cluster computing
  - Special MPI implementation for grid environments
  - Supports usage of different network devices for inter- and intra-cluster communication
  - Developed at RWTH Aachen

Design

Dividing the Problem

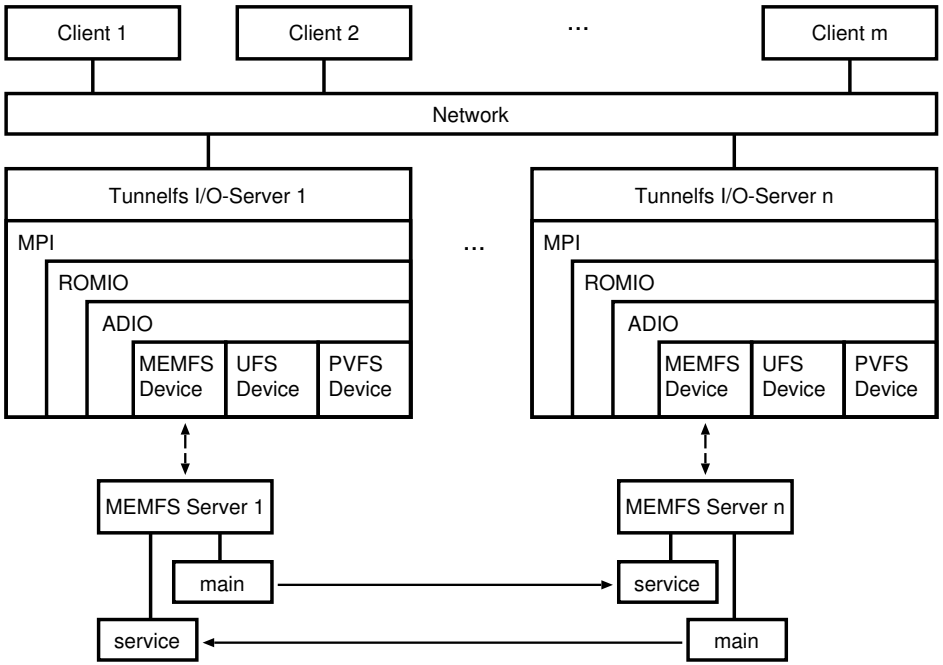
- 2 ADIO devices for high performance remote parallel I/O
- TUNNELFS
  - Transparent access to remote data in a grid
  - Detailed paper was presented at Europar in August 2006 in Dresden, Germany
- MEMFS
  - Data storage in the main memory of local or remote nodes
  - Creates a parallel file system on an arbitrary number of nodes in the grid



Parallel File System

- Simple, deadlock-free communication protocol for data exchange between server nodes
  - All data exchanged with MPI calls
  - MP-MPICH, which is based on MPICH-1 is not threadsafe

→ ensure mutual exclusion of all MPI calls

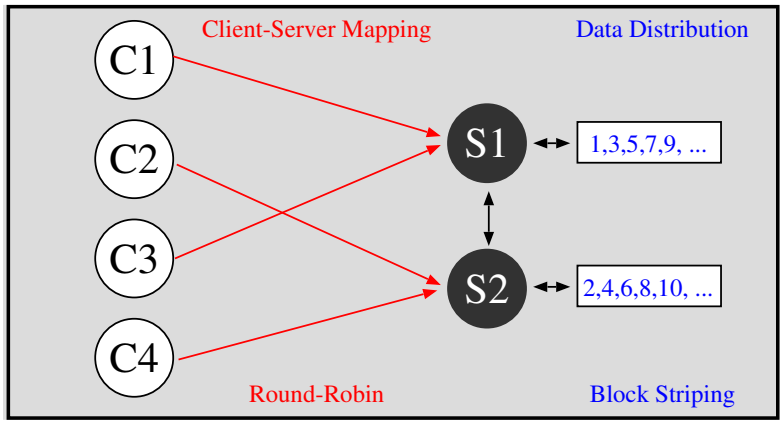


MEMFS File System

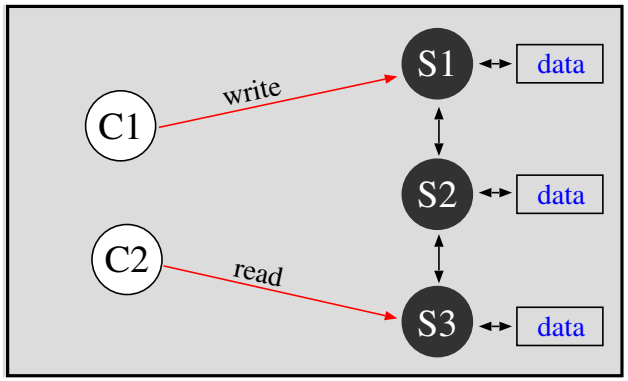
- Calls to ADIO device functions create and manipulate files in the main memory of I/O server nodes
- Dynamic allocation of data blocks
- Non-persistent storage of temporary files
- Total file system size limited by accumulated main memory of I/O server nodes
- Optimized for few, large, temporary files accessed by many clients in parallel
  - Simple directory structure
  - Large block sizes

Client-Server Mapping and Data Distribution

- Client-Server Mapping
  - I/O client contacts one specific server for MPI-IO operations
  - Currently MEMFS uses a static *round robin* mapping to parallelize requests
- Data Distribution
  - Files are distributed among servers to support parallelism between requests from different clients
  - Data distributed by simple striping (variable stripe size)



File Locking

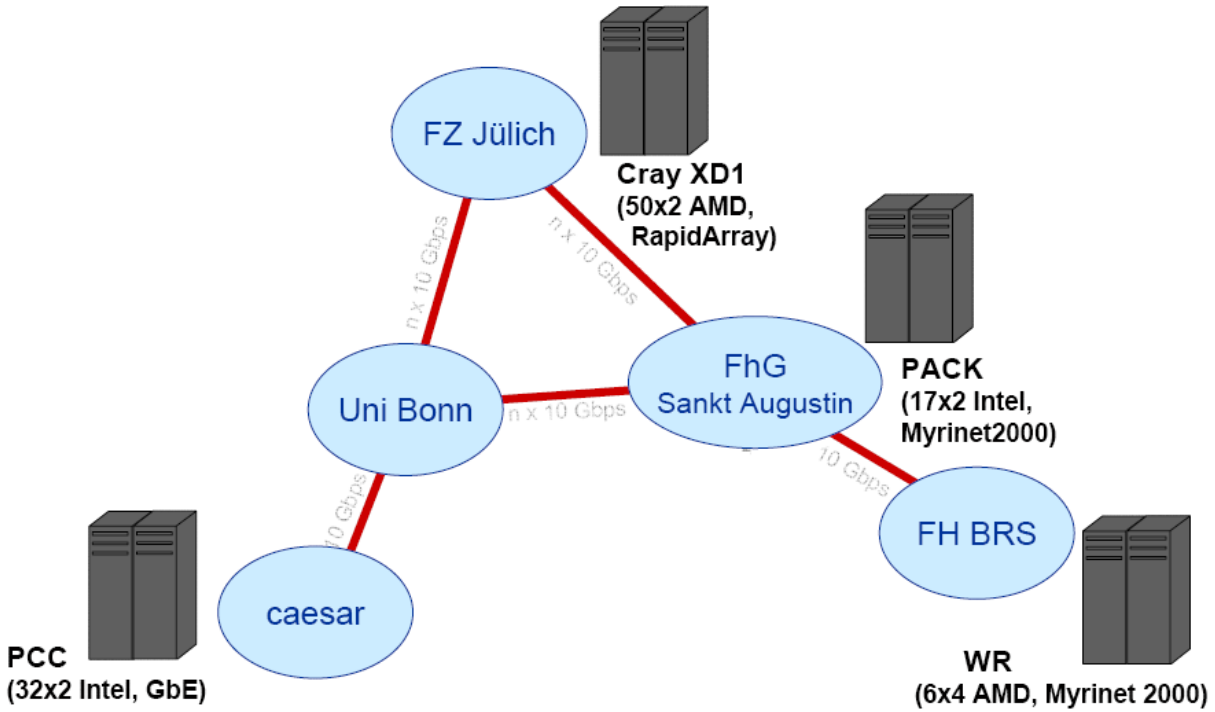


- Several conditions under which sequential consistency has to be guaranteed
- Write accesses to overlapping file regions need to be serialized
- MEMFS uses a centralized block-range locking mechanism
  - Multiple read requests can be performed concurrently
  - User can specify if cost-intensive locking mechanism should be disabled

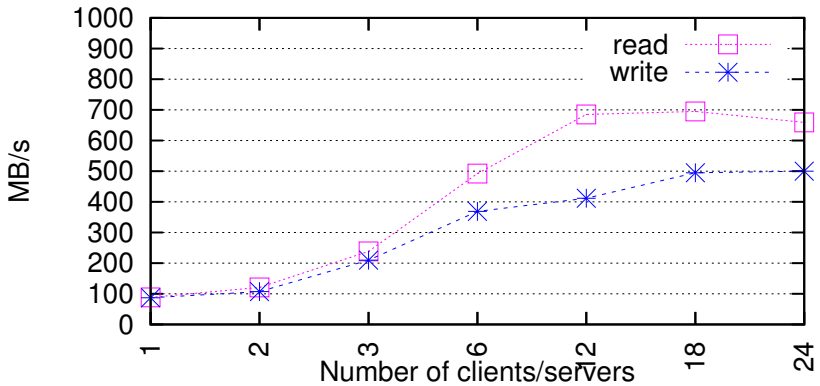
Results

Evaluation Setup

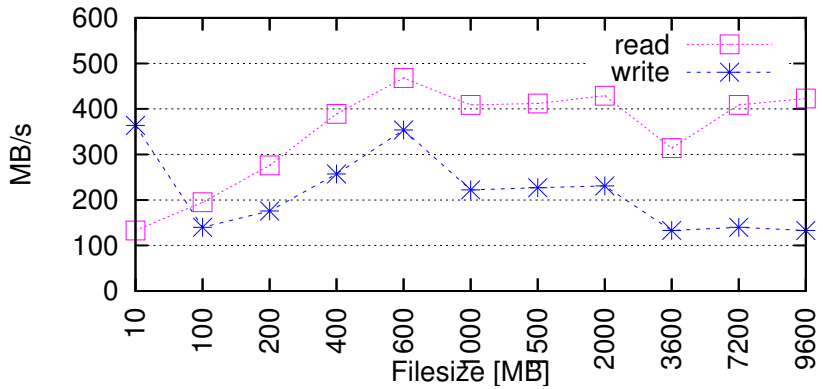
- MEMFS and TUNNELFS were evaluated by measuring the I/O performance between clusters in the VIOLA network
- Benchmark program that performs standard MPI-IO functions
- Clients access disjunct file regions



Results



Varying number of clients and servers



Varying file size. Number of clients and servers fixed to 6

- Good read throughput on large chunks
  - Up to 5.6 of 6 GBit/s reached with 100 MB per client read in a single MPI-IO call
- Optimizations required for more complex access schemes
  - Multidimensional data partitioning between clients (file views)
  - different (small) chunk sizes

Conclusion

Conclusion and Future Work

- Conclusion
  - 2 ADIO devices for high-performance remote parallel I/O to main memory
  - Shows good performance for simple access schemes
  - Evaluated coupling of up to 4 clusters
- Future Work
  - Increase performance for typical access schemes of scientific applications
  - Evaluation of sophisticated client-server mapping algorithms
    - Minimize server-to-server communication for data requests
  - Combined with a suitable data distribution scheme
  - Extension to use MEMFS as a caching system for high-bandwidth I/O during application runtime