Service Challenge 3
Plans @ CERN

Vladimír Bahyl
IT/FIO

Vladimir.Bahyl@cern.ch

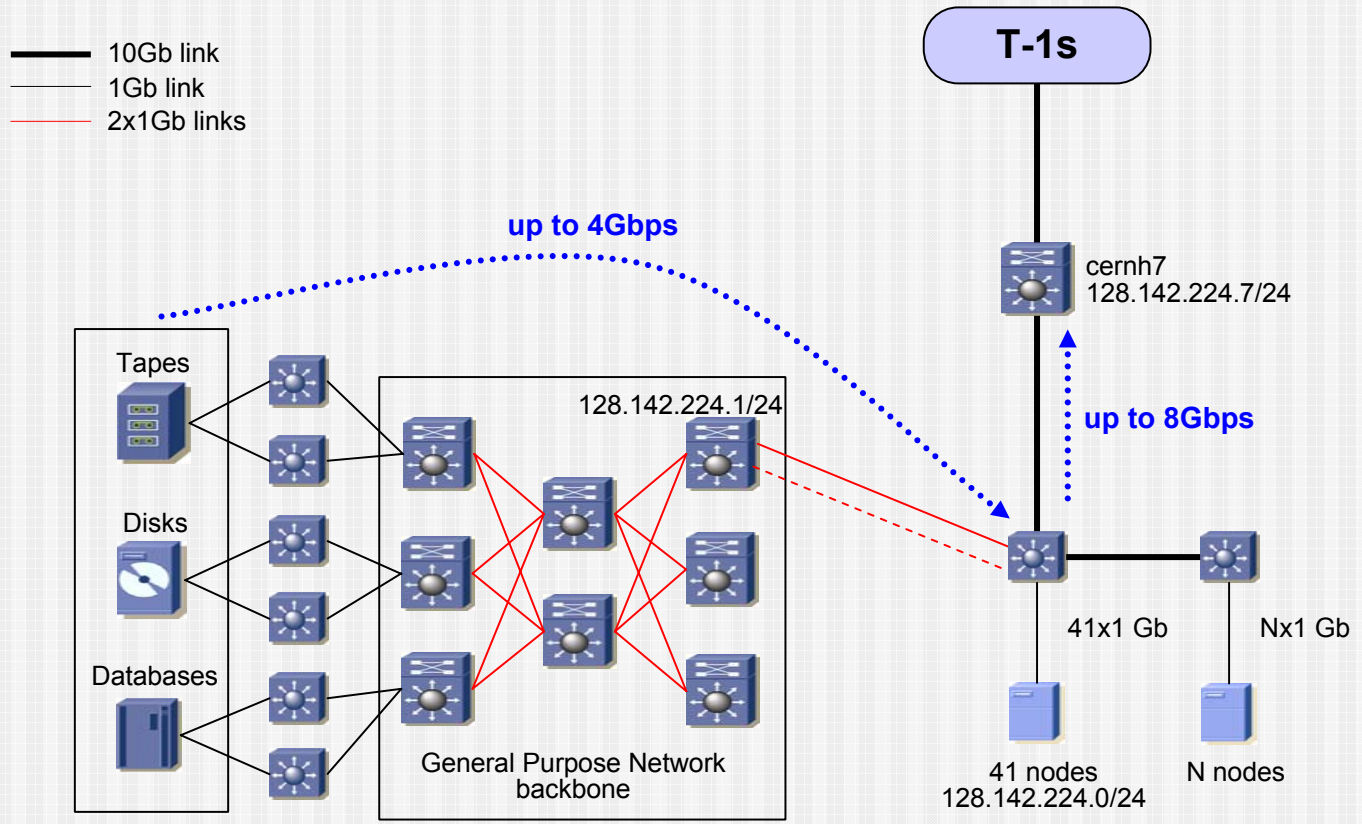
Outline

- Important dates
- Network connectivity
- Hardware
- Software
  - 4 different configurations
- Conclusion

Important dates

- Setup phase
  - Starts *1st of July* 2005
    - But preparations already started
  - Includes throughput test
  - 150 MB/s disk (CERN) → disk (Tier-1)
  - 60 MB/s disk (CERN) → tape (Tier-1)
  - = traffic up to 1 GB/sec from CERN
- Service phase
  - Starts in *September* → end of 2005
  - Includes real experiments' data
    - CMS and ALICE in the beginning
    - ATLAS and LHCb join later (October/November)

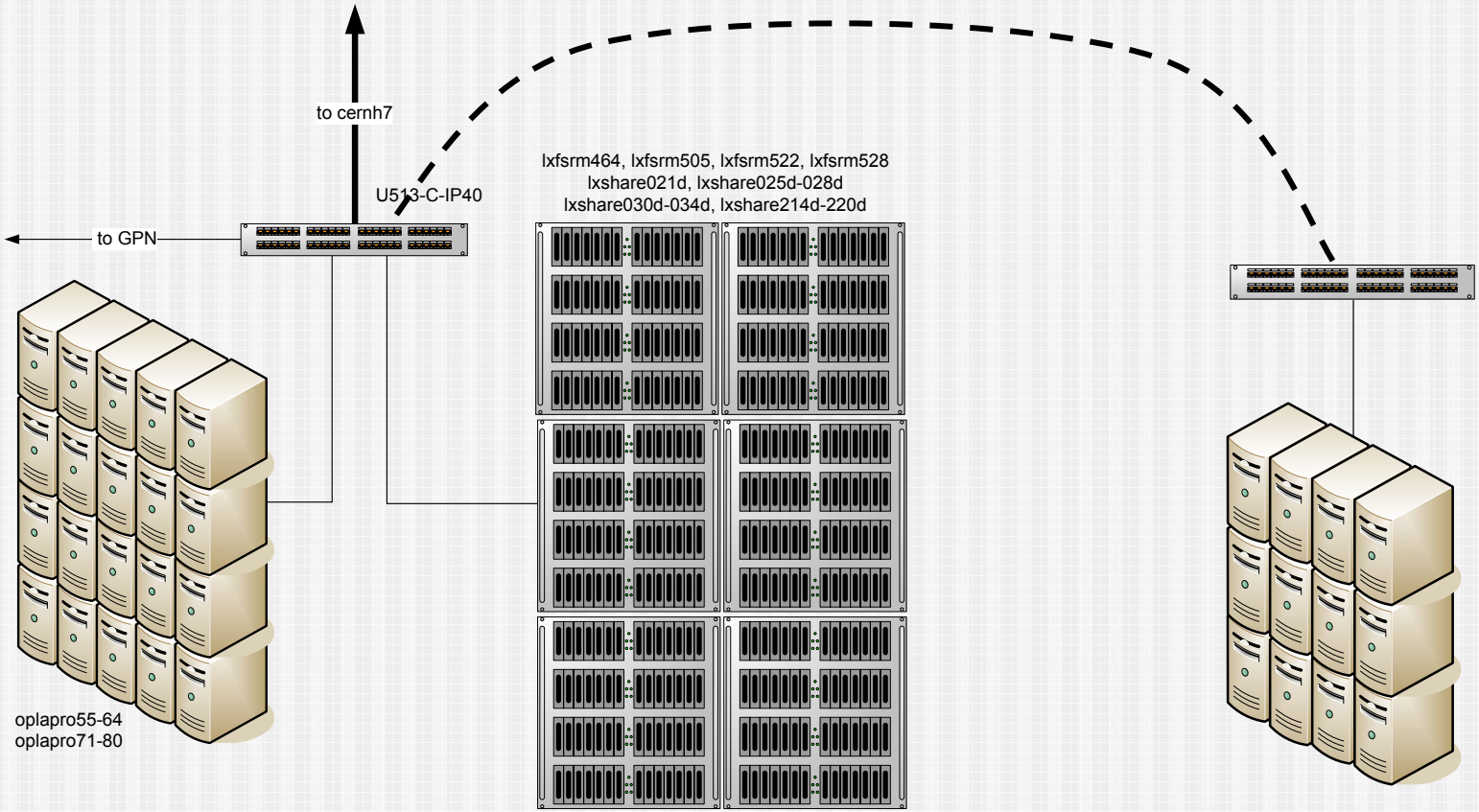
Network connectivity



Hardware – list

- 20 IA64 Itaniums
  - oplaproXX
- 21 IA32 disk servers
  - lxshareXXXd
- 10 other IA32 servers available on standby
  - lxfsXXXX / lxbXXXX
- All nodes will use single Gb network connection only
- Special routing to Tier-1s
- Local host firewall enabled
  - iptables

Hardware – layout



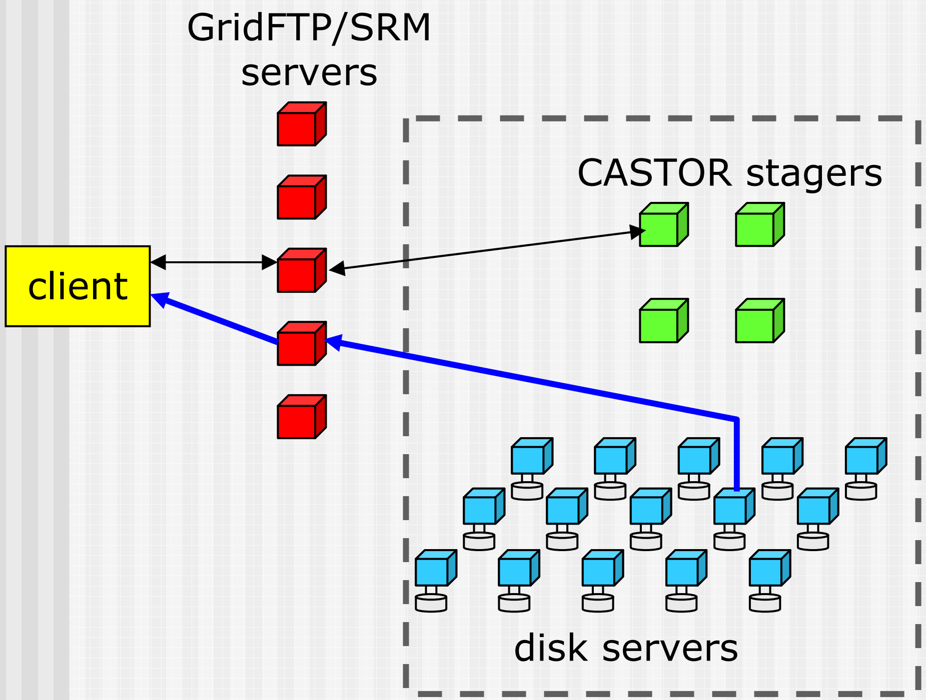
Software – July 2005

- Setup phase with throughput test
- Data from local disks
- SRM + GridFTP for transfers

Software – September 2005

- Service phase – experiments' real data will be transferred
- Currently considering 4 different configuration scenarios
  - http://cern.ch/service-radiant/cern/plans-for-SC3.html

Software – scenario 1



Connection process:

1. Client will connect to any one of the SRM servers
2. SRM server will contact one of the CASTOR stagers to enquire about the location of the data and send it back to the client
3. Client will then connect to the GridFTP server on the same DNS load balancing alias and ask for the data
4. GridFTP server will pull data from the disk server and serve it back to the client

Pros:

☺ GridFTP/SRM separated from the rest of CASTOR services

☺ Easy to expand by adding more GridFTP/SRM nodes

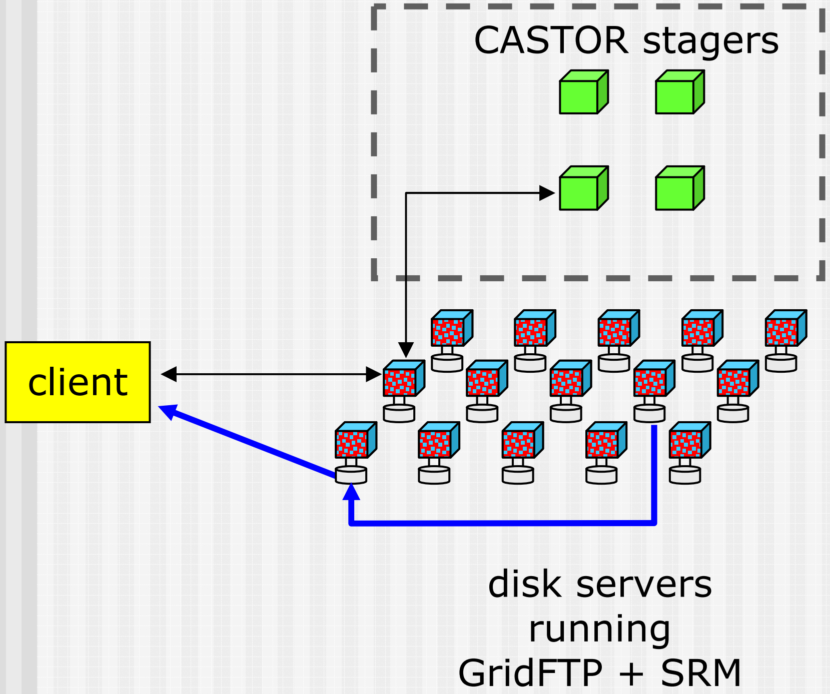
Cons:

☹ Current SRM limitation – bulk transfers served by the same GridFTP node

☹ Limitation on the number of files in the CASTOR stager

☹ Inefficient data transfers (twice via the same node)

Software – scenario 2



Connection process:

1. Client talks to SRM to ask for a disk server (like at CERN)
2. SRM will ask stager about the location of the data and send this information back to the client
3. Client will connect to a disk server running GridFTP daemon (using DNS load balancing alias)
4. In the ideal case, where data is available locally on the disk server, it will be sent back to the client via GridFTP
5. If the data is not local, GridFTP daemon running on the disk server will then pull it from the other disk server that has it and transfer it to the client

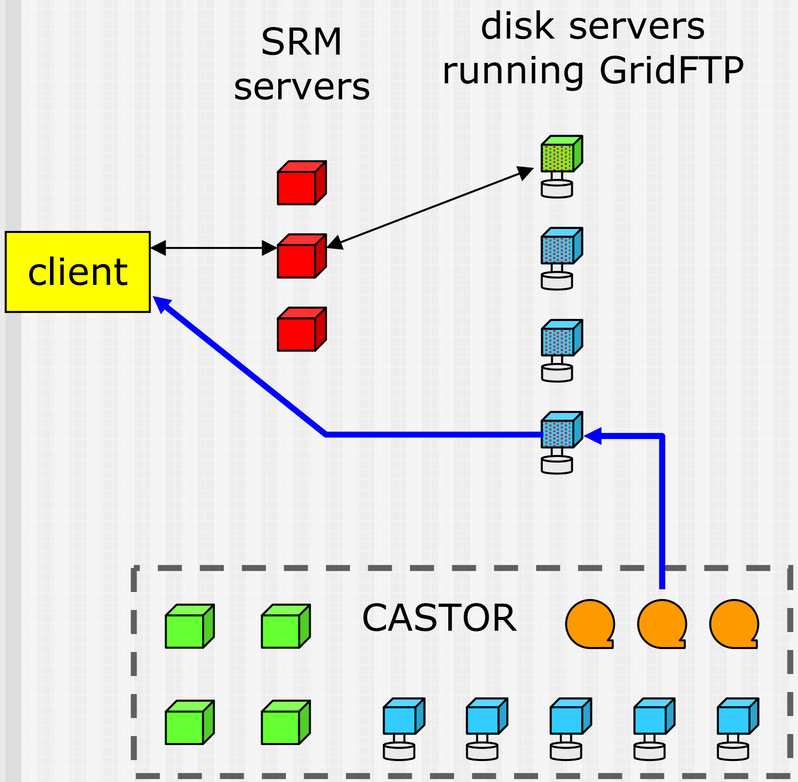
Pros:

- ☺ Data travels to client directly from a disk server (ideal case)
- ☺ Setup will easily support 1 GB/s or more

Cons:

- ☹ Currently, GridFTP can not deal with locality of the data (2006 ?) – until then data will have to be transferred between disk servers
- ☹ GridFTP is a heavy process to run on already loaded disk servers
- ☹ All disk servers would need to have their network settings changed (major operation)
- ☹ Limitation on the number of files with the current CASTOR stager

Software – scenario 3



Connection process:

1. Client at Tier-1 will contact one of the front-end SRM servers
2. SRM will ask local stager about the location of the data
3. If the data is not available on any of the attached disk servers, stager would stage it in from tape
4. SRM will send the information about the location of the data back to the client
5. Client will connect to the given disk server and fetch the data over GridFTP

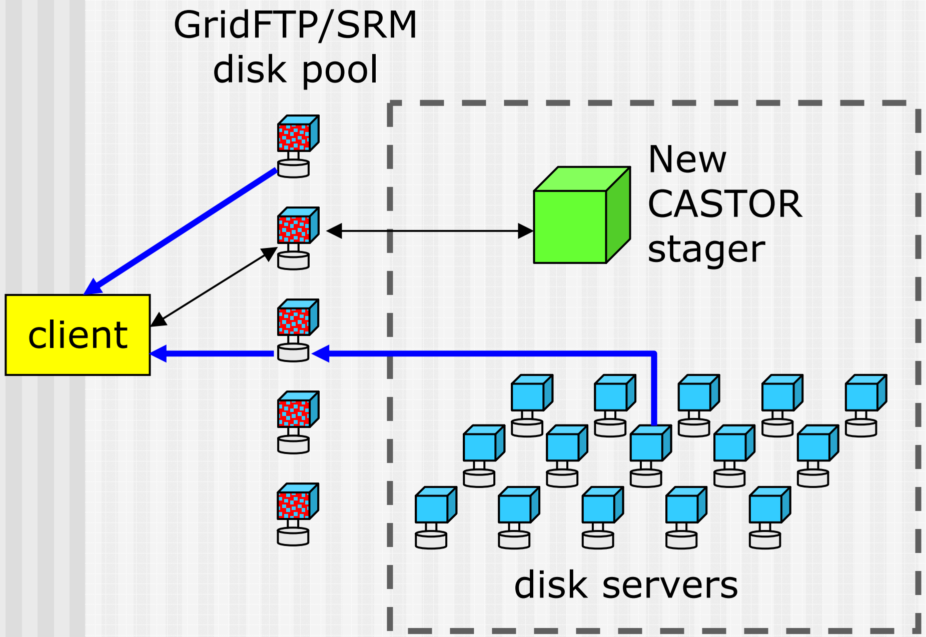
Pros:

- ☺ Elegant separation of services = do not interfere with the rest of CASTOR
- ☺ Easily expandable by adding more nodes
- ☺ Should support rates of 1 GB/s or more

Cons:

- ☹ Data would have to be migrated to tape first = impossible to read data generated on the fly
- ☹ (Pre-)staging of existing data required
- ☹ SRM would have to be modified to return TURL with disk server name
- ☹ Limitation on number of files with the current CASTOR stager
- ☹ Thrown away solution in general

Software – scenario 4



Connection process:

1. Client at Tier-1 will request a file from one of the externally visible front-end disk servers running SRM and GridFTP
2. If the data is present in the disk pool, it's location is forwarded to the client which will then fetch the data via GridFTP from the given disk server
3. If the data has to be fetched from another disk server (inside CERN), new CASTOR stager will replicate it to the GridFTP disk pool of externally visible disk servers
4. Client will be informed about the location of the data and fetch it from appropriate disk server via GridFTP

Pros:

- ☺ No limitation on the number of files in the CASTOR catalog
- ☺ Easily expandable by adding more nodes to the GridFTP disk pool
- ☺ Should support rates of 1GB/s or more
- ☺ Few nodes exposed externally
- ☺ Separation of WAN services from CERN internal CASTOR operations
- ☺ Closest to the real LHC scenario

Cons:

- ☹ Requires lot of new software to be in place within < 2 months
- ☹ In the worst case, every file would have to be replicated between 2 disk servers = load on 2 nodes

Conclusion

- Completed:
  - Network setup is ready
  - Hardware configuration of machines
  - Nodes installed with Scientific Linux CERN 3.0.4
    - IA32 fully Quattorized
- In progress:
  - Quattorize IA64 nodes
  - Finalize GridFTP/SRM/CASTOR stager configuration
  - Deployment of the new CASTOR stager
  - SRM development
- To do:
  - Decide which option we take and implement it

Thank you

- E-mail: Vladimir.Bahyl@cern.ch