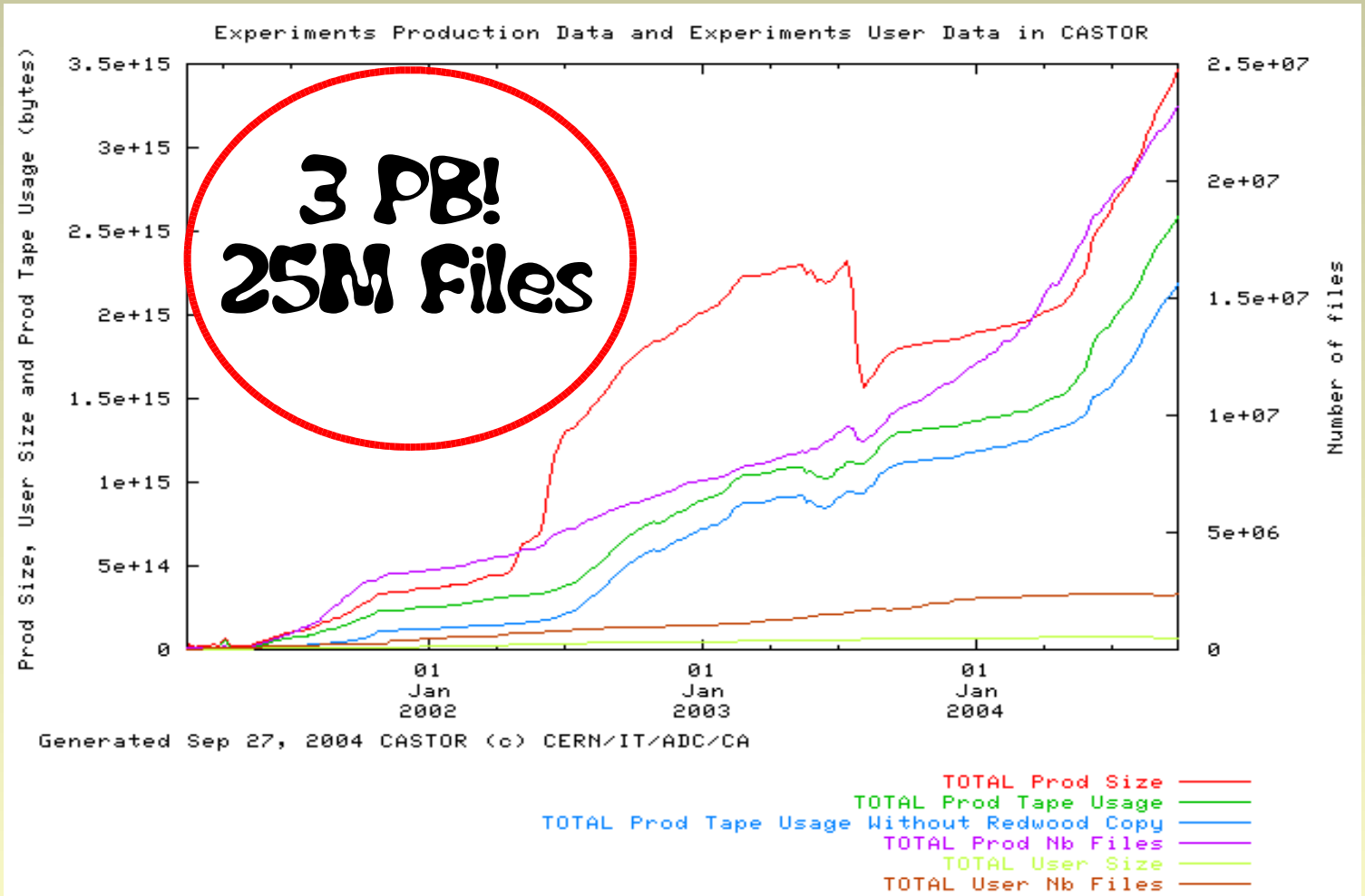
CASTOR: Operational issues and new Developments

- CASTOR Current State
  - Statistics
  - Setup, Architecture and associated problems
- New CASTOR Stager
  - Challenge, New Architecture, Tests
- Management

Statistics



3 PB!
25M Files

Great, but...

Setup, Architecture...


TAPE SERVER


TAPE SERVER


TAPE SERVER

CENTRAL SERVICES

~90 Tape Drives


~370 Disk Servers for ~300TB

~50 Stagers

STAGER

DISK SERVER

FS FS FS FS

DISK SERVER

FS FS FS FS

DISK SERVER

FS FS FS FS

... and associated problems

- Management more and more difficult
- Order of magnitude increases, and we see:
  - Performance hickups
  - Scalability problems
  - Needs of a proper Resource sharing
  - Some internal limitations become showstoppers
    - Internal catalog
    - Sub optimal use of resource
    - Not scalable
    - ...

Our Challenge

- LHC is our **Challenge**

- A single stager should scale up to handle **500/1000 requests/second (** a 'request' is a file opening)

- **4 PB** of disk cache, **10 PB** on tape/year
- Peak rate of **4 GB/s** from online
- **10000** disks, **150** tape drives
- Increase of small files... and the number of them

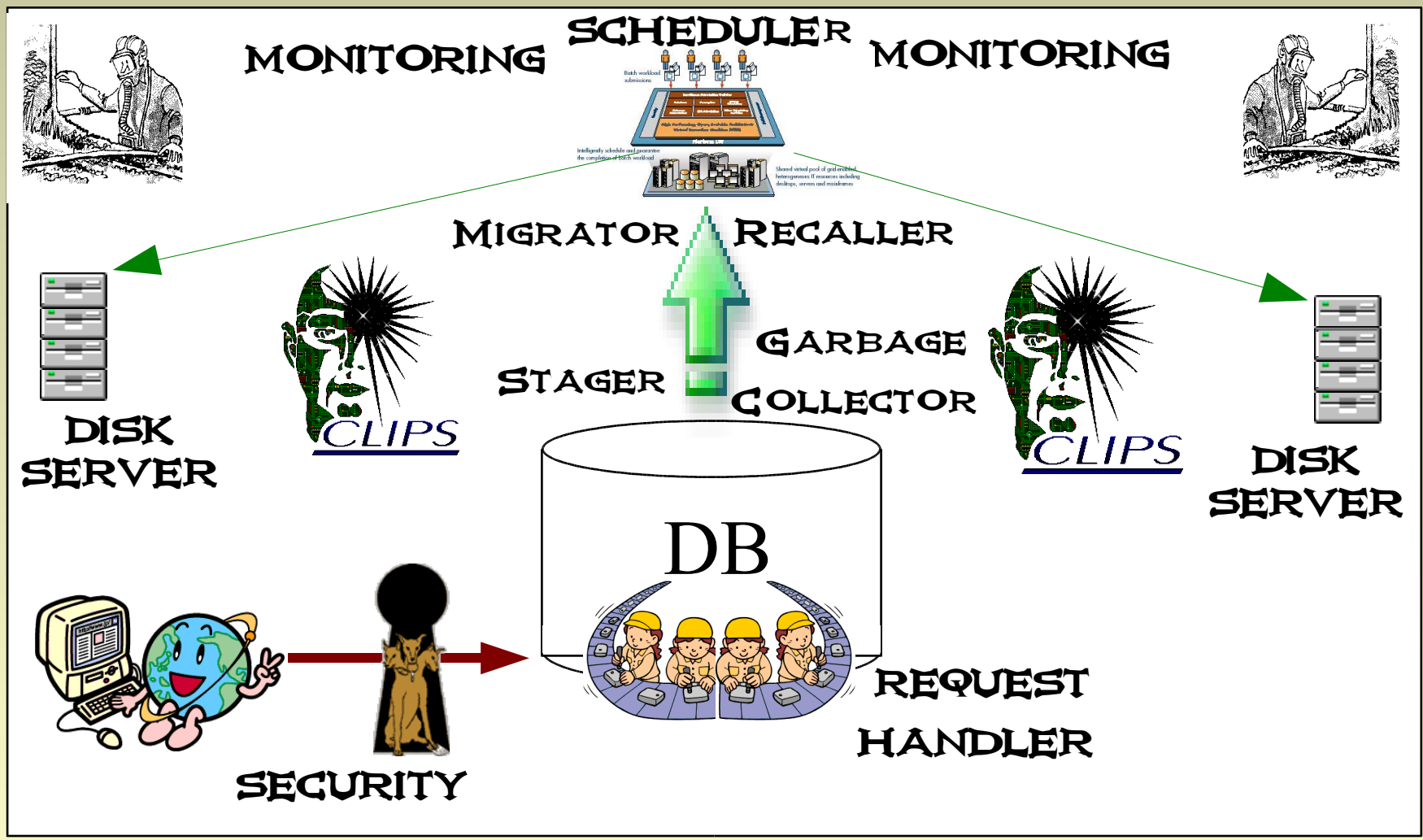
- The current stager cannot do it
  - Conceptually the whole design has to be revisited

Our Vision

- With clusters of 100s of disks and disk servers, the automated management faces more and more the same problems as *Computing* (not storage) resource sharing for CPU clusters
  - Resource management, Sharing, Scheduling
  - Configuration, Monitoring

Storage Resource Sharing Facility

New Architecture



Security

- We are implementing **strong authentication** (encryption is not planned for the moment)

- We have developped a plugin system, based on the GSSAPI so as to use the following mechanisms:
  - **GSI, KRB5**
- And support KRB4 for back compatibility

- We are currently modifying the various CASTOR components to integrate the security layer

- Impact on the **config** of the machines (need for **service keys** etc...)

Tests

- Using an external scheduler
  - proof of concept 1 year ago
  - LSF, MAUI CPU schedulers + our plugin

- Using filesystems not tuned for our application (!?), we can test how it reacts in a non-optimal environment (!)

- We prepare stress tests in a environment similar to the Alice Data Challenge'04.

Test of migration (1)

- Full chain: memory + network + disk + tape

We use 5 disk servers, eaching running xfs, disk array

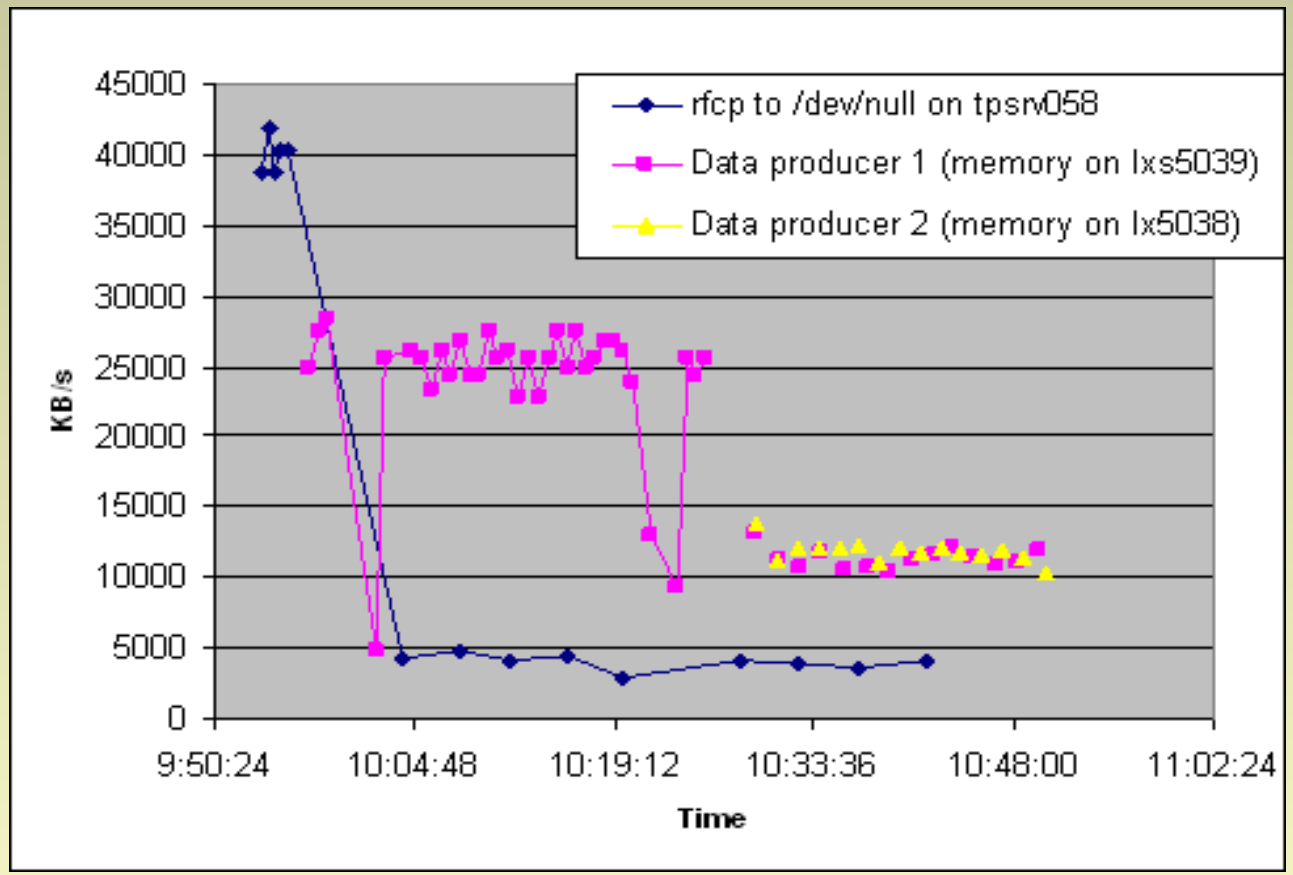
- Environment will not be in favour of CASTOR;
- Filesystems tuned for write, low performance on read if there is a writer
- Nominal 'filesystem' speed about 25-30 MB/s for one remote writer (RFIO / Remote File IO protocol)
- reader speed nominal if no other stream, otherwise 5-10 MB/s

- We schedule 6 write-to-disk producers
  - Almost always 1 producer per machine

Test of migration (2)

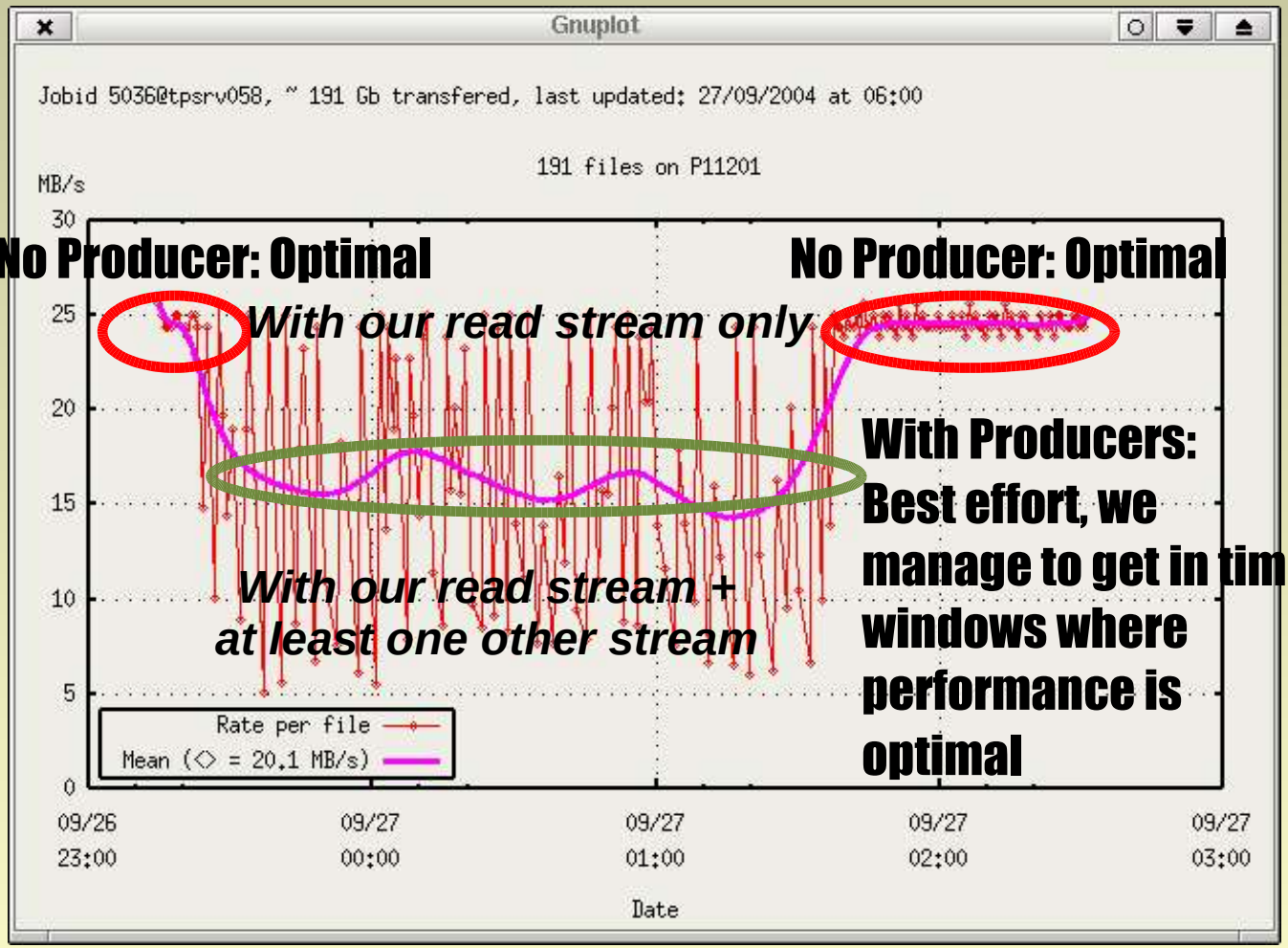
- We add two read-from-disk migrators, to 9940B drives
  - Nominal tape drive speed about 25-30 MB/s
- Files migrated from disk to tape are selected just-in-time when the tape mover is ready to receive more data.
- Selection decision for files being migrated uses scheduling, based on metrics such as CPU, load, etc...
- How will react CASTOR with both producers and readers?

Test of migration (3)



Let's Use xfs on a disk array but tuned for write (sub optimal for CASTOR)

Test of migration (4)



Test of migration (5)

- We perform nominally in the idle case (ouf!)
- We perform better than expected in the bad (e.g. with producers) case by using the small time windows whenever possible
- What next: verify with filesystems tuned for our application
  - We think that a better fs is, if there are one write and one read stream, the read stream should perform as the write stream

Management

- Our operational team took over 100% of CASTOR service management since 1 year

- Adaptation of **LEMON** and **QUATTOR**
  - **LEMON** : Monitoring metrics being added to handle our s/w survey + specific h/w survey (tape drives)
  - **QUATTOR** : ideal for
    - maintaining the SW in synch across the machines
    - maintaining and distributing consistent configuration information from central definitions

- Please refer to other CERN talks (Tim Smith, German Cancio)

Conclusion

- Hybrid (some old pieces left) stager prototype ready for ALICE MDC Performance
  - **450** MB/s aggregate
- Final new stager system: design ready and the implementation of the remaining components of the central framework has started

http://cern.ch/castor