

## Roles:

► Monte Carlo Production ► Analysis

## CPU:

1MSi2k  
~ 500 batch slot

## Storage:

Size: ~ 200TB

WAN transfer (T1<->T2):

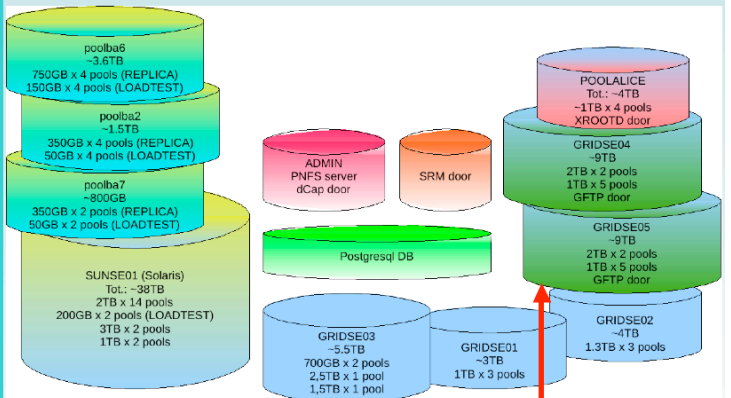
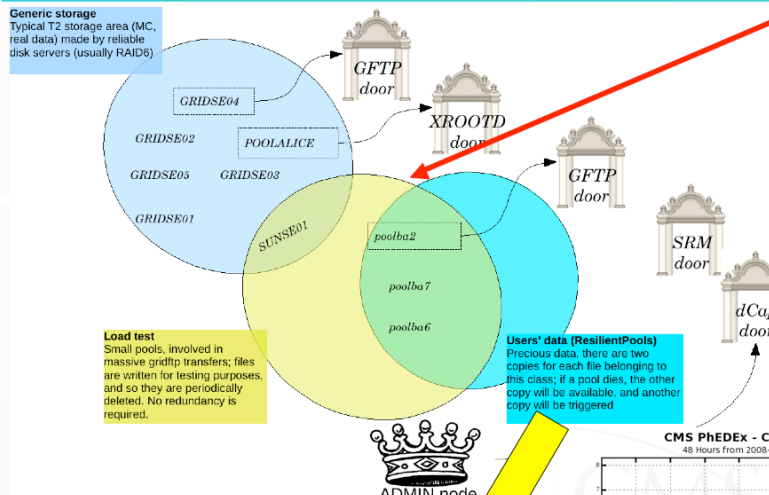
- > 50MB/sec import
- > 10 MB/sec export

LAN transfer (Storage<->WN):

- ~ 1GB/sec aggregate read from WNs
- > 10 MB/sec write to storage

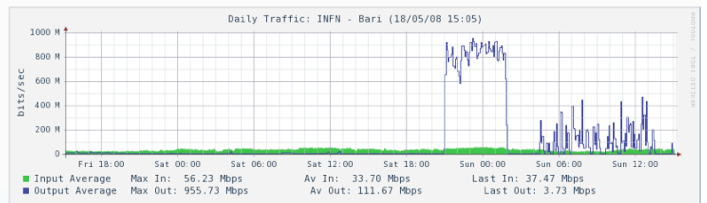
Interactive data read/write access from the user (mainly belonging to the "Tier2 local community")

Size is small compared to other kind of data but requires low latency access

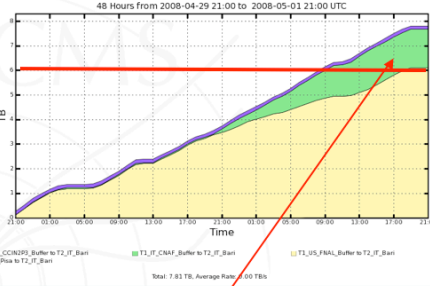


## Typical scenario

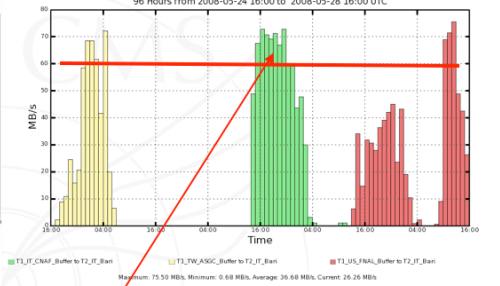
- More than one VO per site
- Several and different storage "boxes"
- Different "kind" of data with different usage pattern and "retention policy"
- "short living" and test data
- "MC produced data"
- Experimental/MC data (replica of T1 data)
- User's data



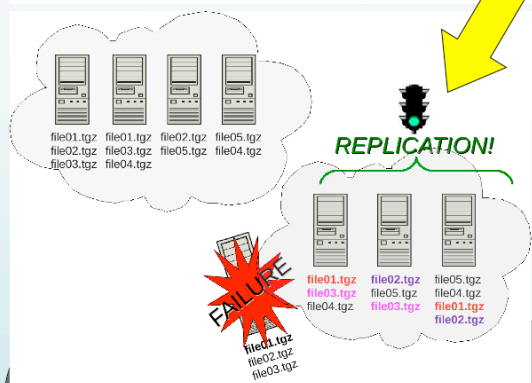
CMS PhEDEx - Cumulative Transfer Volume



CMS PhEDEx - Transfer Rate



Already achieved the goal on the WAN activity of a typical Tier2 during LHC run



- In the case of many parallel access we see better performances when disks are used independently instead of in RAID
- We exploit this configuration in order to achieve good performance and good availability using ReplicaManager
- We tested several OS and kernels and found that "Debian stable" seems reliable and fast enough for our purpose.
- We tested successfully "deadline" scheduler of the linux kernel (>2.6.18)
- We tested successfully "Thumper" (Fire 4500) with Solaris+ZFS:
  - It provides really high performance
  - It is easy to set-up and configure (network trunk and ZRAID)
  - It provides good reliability and great manageability

## OS optimization

- Setting properly the "blockdev" in the kernel improved the I/O performance
- Setting properly the "nr\_requests" in the kernel improved the ability of the machine to schedule the I/O command

STRESS TEST: Writing 30 files while reading 30 other files

READ SPEED			WRITE SPEED		
BLOCKDEV(*)	RAID 6 HW	RAID 6 SW	BLOCKDEV(*)	RAID 6 HW	RAID 6 SW
1024	30MB/s	16MB/s	1024	21MB/s	21MB/s
16384	50MB/s	19MB/s	16384	55MB/s	21MB/s
65536	31MB/s	18MB/s	65536	32MB/s	21MB/s

(\*) readahead in 512-byte sectors: 16384~8KB