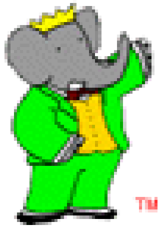
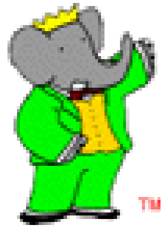


BaBar @ CC-IN2P3



An example of data management in a Tier A/1

Jean-Yves Nief

CC-IN2P3

GDB meeting - Lyon - 16/03/05

Overview of BaBar @ CC-IN2P3

- CC-IN2P3: « mirror » site of Slac for BaBar since November 2001:
  - real data.
  - simulation data.

  (total = 290 TB: Objectivity eventstore (obsolete) + ROOT eventstore (new data model))
- Provides the services needed to analyze these data by all the BaBar physicists ***(data access)***.
- Provides the data in Lyon within 24/48 hours after production ***(data management)***.
- Provides resources for the ***simulation production***.

Data access.

Overview of BaBar @ CC-IN2P3 (general considerations)

- Large volume of data (100's TB).
- Mainly, non modified data(write once, read many times).
- Number of clients accessing the data in // (100's to 1000's):
  - Performant access necessary: **Latency time reduced.**
  - Data volume / demand increasing over time: **Scalability.**
- Using distributed architectures:
  - **Fault tolerance.**
- Hybrid storage (tapes + disk):
  - **Transparent access to the data on tapes.**
  - **Transparent disk cache management.**

BaBar usage @ CC-IN2P3

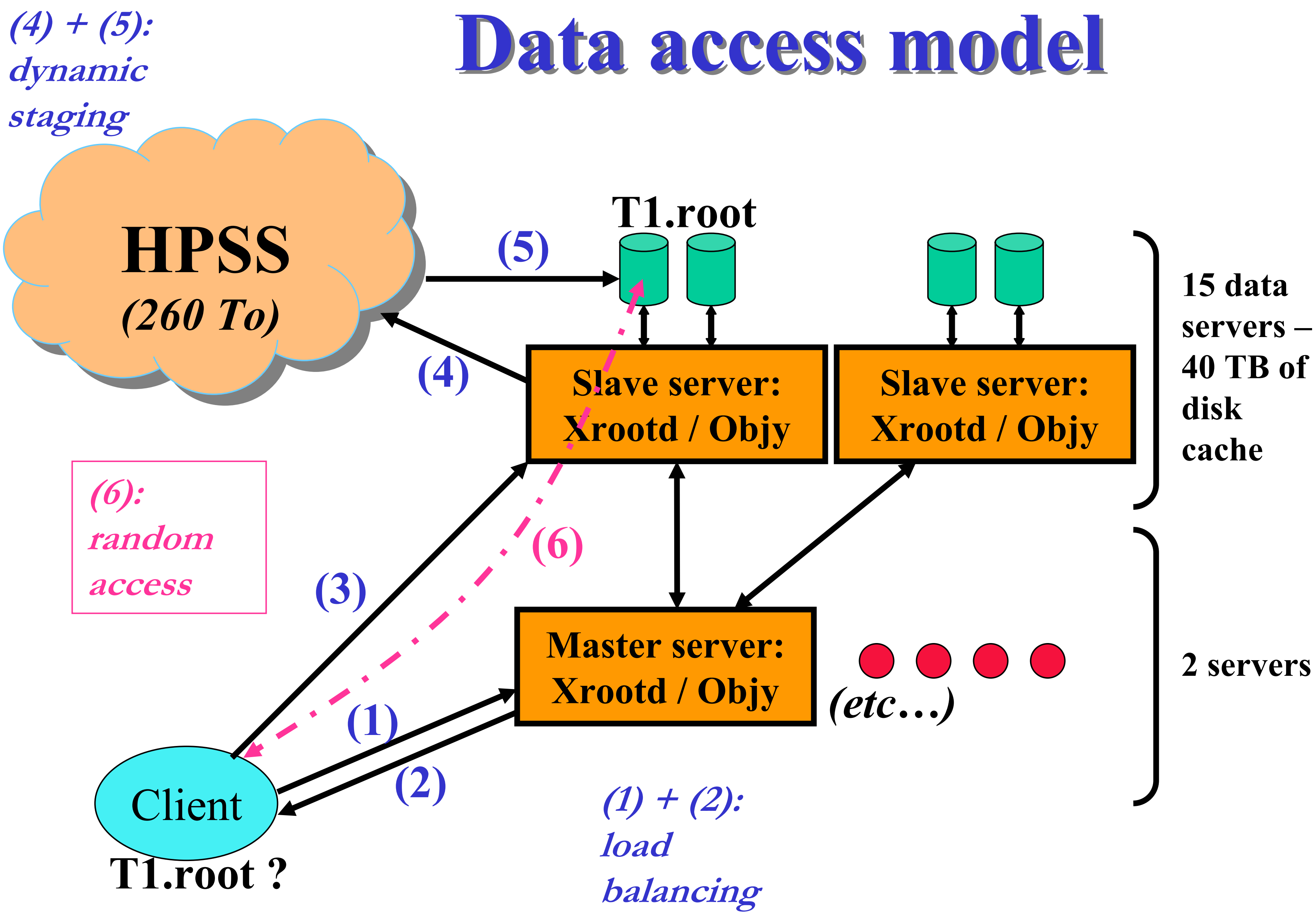
- 2002 – 2004: **~ 20-30%** of the CPU available.
- Up to **600** users' jobs running in //.
- « Distant access » of the Objy and root files from the batch worker (BW):

  ➔**random access** to the files: only the objects needed by the client are transfered to the BW (~kB per request).

  ➔ hundreds of connections per server.

  ➔ thousands of requests per second.

Data access model



Xrootd for the data access (I).

- Scalable.
- Very performant (trash NFS!).
- **Fault tolerant** (server failures don't prevent the service to continue).
- Lots of freedom in the site configuration:
  - Choice of the hardware, OS.
  - Choice of the MSS (or no MSS), protocol being used for the dialog MSS/Xrootd (ex: RFIO in Lyon).
  - Choice of the architecture (ex: proxy services).

Xrootd for the data access (II).

- Already being used in Lyon by other experiments (ROOT framework):
  - D0 (HEP).
  - INDRA (Nuclear Physics).
  - AMS (astroparticle).
- Can be used outside the ROOT framework (POSIX client interface):

  Ex: could be used in Astrophysics for example (FITS files).

➔ **Xrootd: very good candidate for data access at the PetaByte scale.**

BUT….

Data structure: the fear factor (I).

- A performant data access model depends also on this.
- Deep copies (full copy of subsets of the entire data sample) vs « pointers' » files (only containing pointers to other files) ?

Deep copies	« Pointers » files
- duplicated data	- no data duplication
- ok in a «full disk» scenario	- ok in a «full disk» scenario
- ok if used with a MSS (if not too many deep copies!)	- potentially very stressful on the MSS **(VERY BAD)**

Data structure: the fear factor (II).

- In the BaBar Objectivity event store:
  - Usage of pointer skims: very inefficient ➔ people build their own deep copies.
  - For a single physics event:
    - Data spread over several databases.

    ➔ At least 5 files opened (staged) for one event!
- Deep copies are fine, unless there are too many of them!!! ➔ data management more difficult, cost increasing (MSS, disk).

The best data access model can be ruined by a bad data organization.

Dynamic staging.

- What is the right ratio ***ratio (disk cache / tape)*** ?
- Very hard to estimate, no general rules! It depends on:
  – The data organization (« pointers » files? …).
  – The data access pattern (number of files « touched » per jobs, total number of files potentially being « touched » by all the jobs per day).

➔ Estimate measuring (providing files are read more than once):
  - lifetime of the files on the disk cache.
  - time between 2 restaging of the same file.

- Right now for BaBar ***ratio = 44 %*** (for the ROOT format) ➔ OK, could be less (***~30%*** at the time of Objectivity).
- Extreme cases: Eros (Astrophysics) ***ratio = 2.5%*** is OK (studying one area of the sky at a time).

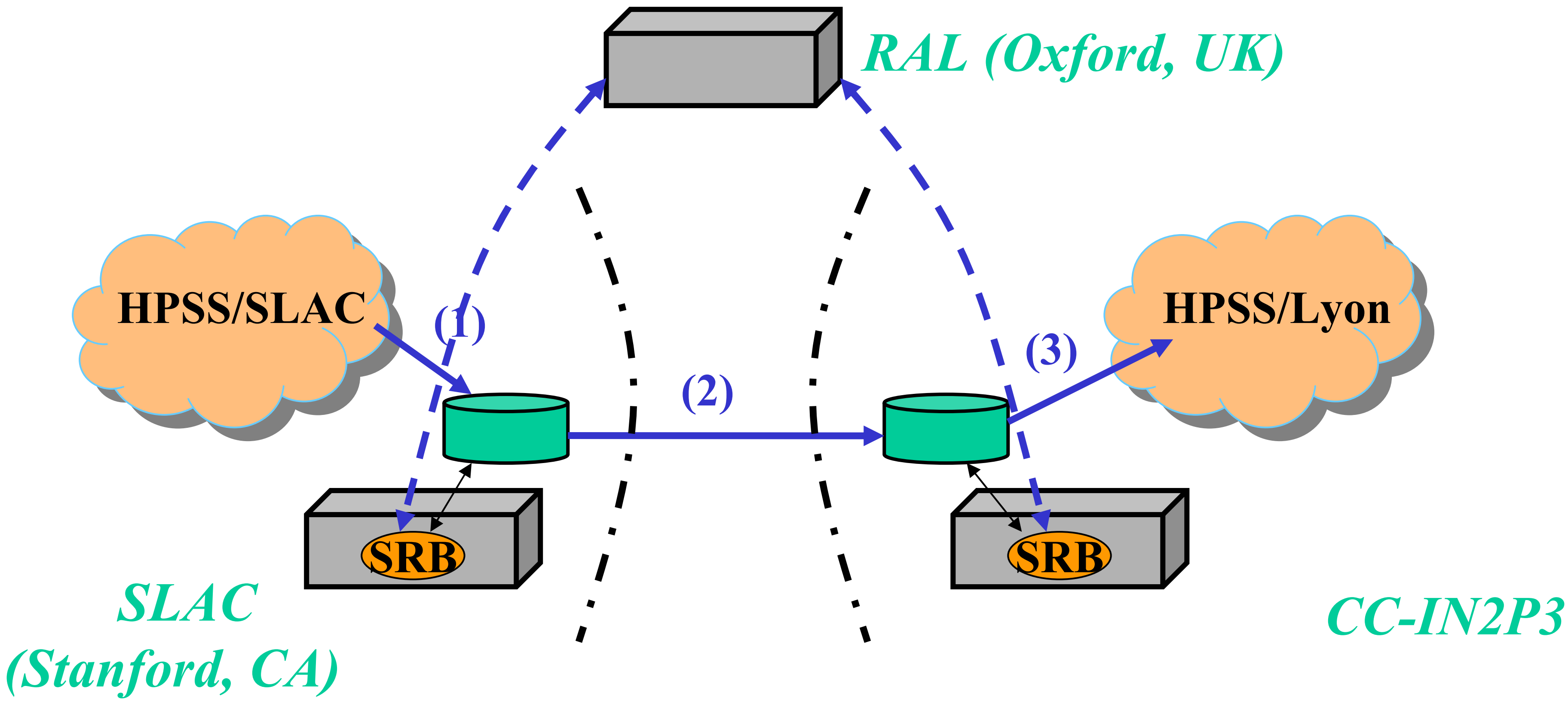
Data management.

Data import to Lyon.

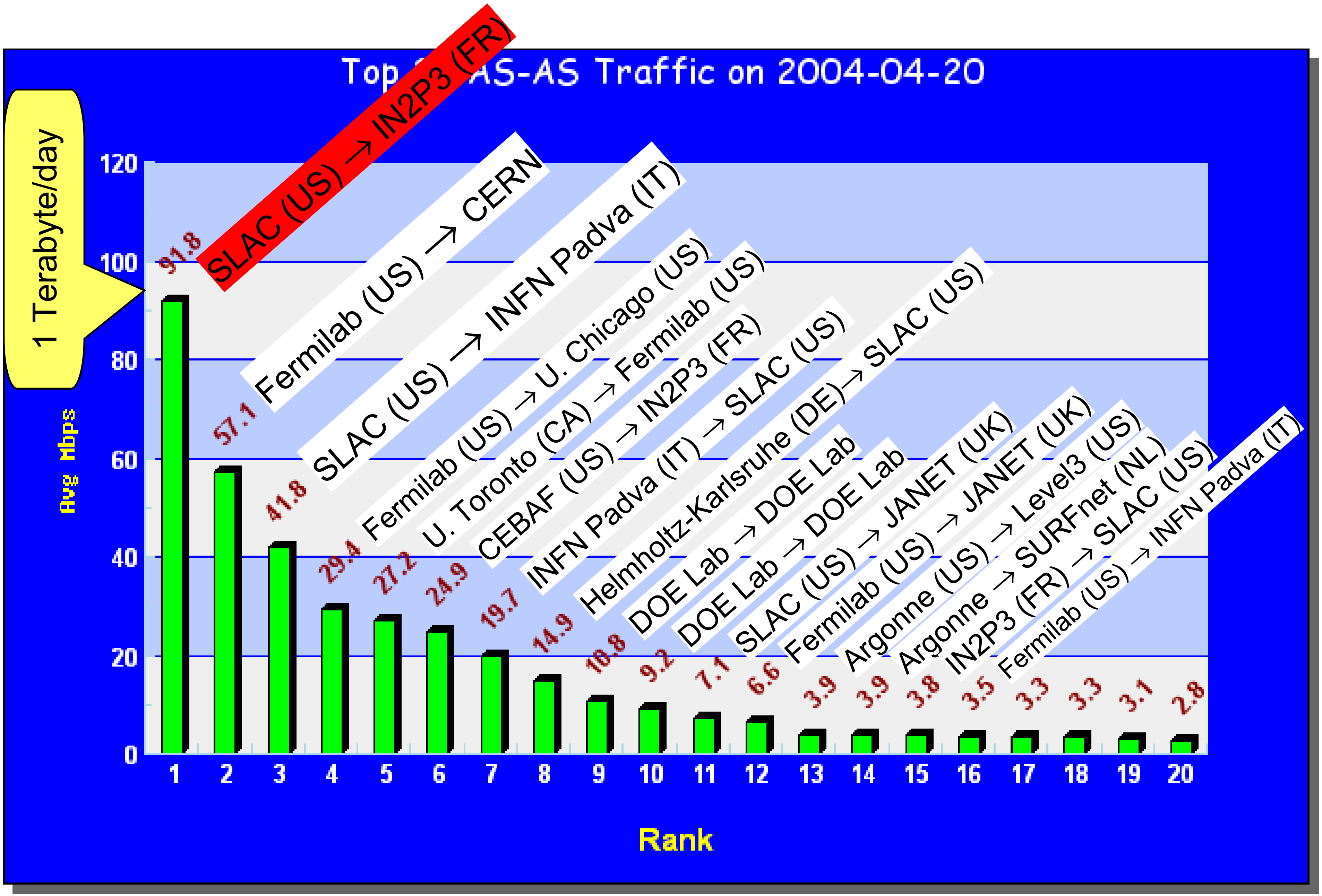
- Data available within 24/48 hours:
  - Hardware: performant network, servers configuration should scale.
  - Software: performant and robust data management tools.
- Since January 2004, using SRB (Storage Resource Broker):
  - Grid middleware developed by SDSC (San Diego, CA).
  - Virtualized interface to heterogeneous storage devices (disk, tape systems, databases).
  - Portable on many platforms (Linux, Solaris, AIX, Mac OS X, Windows).
  - Handling users, access rights, replica, meta data and many, many more.
  - API available in various languages (C, Java, Perl, Python), Web interfaces.
  - Used **in production** in many areas: HEP, biology, Earth science, astrophysics.
  - Save a lot of time in developing performant and automatic applications for data shipment.

SRB and BaBar.

- Massive transfers (170,000 files, **95 TB**).
- Peak rate: **3 TB / day** tape to tape (with 2 servers on both side).



Example of network utilization on ESNET (US): 1 server.



Data import: conclusion.

- SRB:
  - Very powerful tool for data management.
  - Robust and performant.
  - Large community of users in many fields.
- Pitfalls:
  - Huge amount of files to handle.
  - If a some of them missing:

  ➔ Should be easy to track down the missing files:
  Logical **F**ile **N**ame ←→ **P**hysical **F**ile **N**ame (was not the case within Objectivity framework).

  ➔ Good data structure important.

Simulation production.

BaBar simulation production.

- For the last production: SP6.
- More than 20 production sites.
- Data produced at each sites shipped to SLAC and redistributed to the Tier 1.
- CC-IN2P3: 11% of the prod.
- 2nd largest producer.
- but **~ 80%** of the prod in non Tier 1 sites.
- activity completly distributed.
- **Important role of the non Tier 1 sites.**

Conclusion.

1. Data access / structure model: the most important part of the story.
2. Xrootd: very good answer for performant, scalable and robust data access.
3. Interface SRM / Xrootd: valuable for LCG.
4. Ratio (disk space / tape): **very hard to estimate**. Needs at least experience with a test-bed (after having answered to 1.).
5. Data management: SRB great!
6. Lessons learned from past errors: computing model « lighter ».