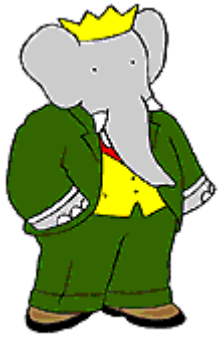
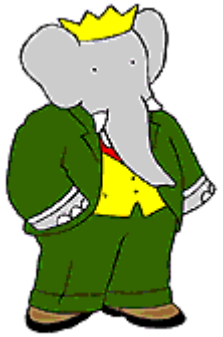


Simulation production in BaBar CM2

Douglas A. Smith
Stanford Linear Accelerator Center
Fred Blanc
University of Colorado
Dirk Hufnagel
University of Ohio

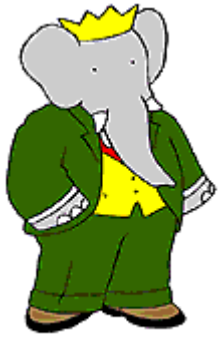
CHEP 2006
Feb. 15, 2006, TIFR, Mumbai, India

For the BaBar computing group



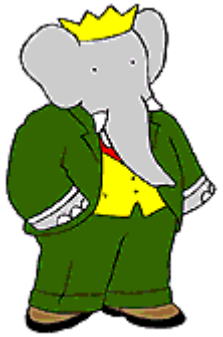
Simulation Production in BaBar

- In current HEP experiments there is a need for a large amount of simulated events to be produced. (In BaBar 3 times data for B-pair, equal to data for continuum.)
- The size and scope of meeting this need now lies beyond the ability of any one computing site: ave. production cycle is ~2M jobs, ~1000 cpu years, ~50TB of data; needed in less than one year for use.
- In BaBar this is now a distributed computing problem, to ~20 sites.
- See D. Smith, *et al.*, Talk 339, CHEP 2004 -- for more details.
- For GRID use see: C. Brew, Poster 106, CHEP 2006.



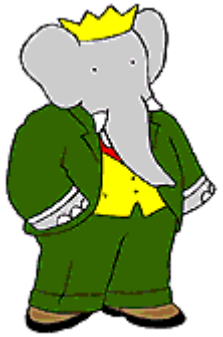
BaBar CM2

- The first computing model included that data storage would be into databases. Was put into use at the beginning around 1998.
- As the experiment continued this was shown to not be scalable to needed effort, a computing model 2 (CM2) plan was started.
- The resulting CM2 included that data storage will be in root files. Developed through 2003 -- put into use in early 2004. See P. Elmer, Talk 502, CHEP 2004 – for more details.
- Going to CM2 was a lot of work.
- But needed, support issues in using a database was holding back the experiment from scaling to the size of computing effort that is currently needed.



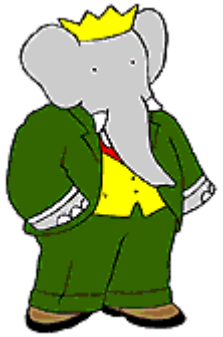
Why to not use a database

- There are specific support issues for data production:
  - All production was into one database, **tying together jobs**. Effecting the database, effected all jobs. Database issues could stop all jobs, exporting data would need to know what jobs were touching which databases.
  - Database **stability could kill jobs**. Hard to diagnose since they were transitory problems, and could not be replicated. This resulted in 4-6% failure rate in production, and could never be made better.
  - Little to no knowledge of **which files contained which data**. Data for an event would be spread among many files, and the data in a file would be spread among many collections of events. Producing problems with data distribution and management.
  - Production managers also needed to be **database admin. experts**.
  - Other items: mysterious cache settings, closing transactions when jobs failed, corruptions effecting large chunks of data, container creation slowing the startup of jobs, etc.



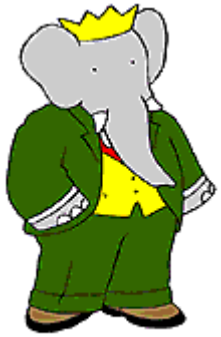
Issues going to root files

- But producing data into root files brought up issues. There are features in database use that we now lose, and need to recreate.
- Databases naturally produced data into **larger chunks**, now each 5-6 hour production job produces a few ~50Mb files.
- Databases included **file management**, and now we would have to manage the produced root files ourselves.
- Because jobs would produce relatively small files, need to **merge output** into larger root files for distribution and use.
- Now merging of data into larger chunks and management of files and export were up to us. **Extra development effort**.

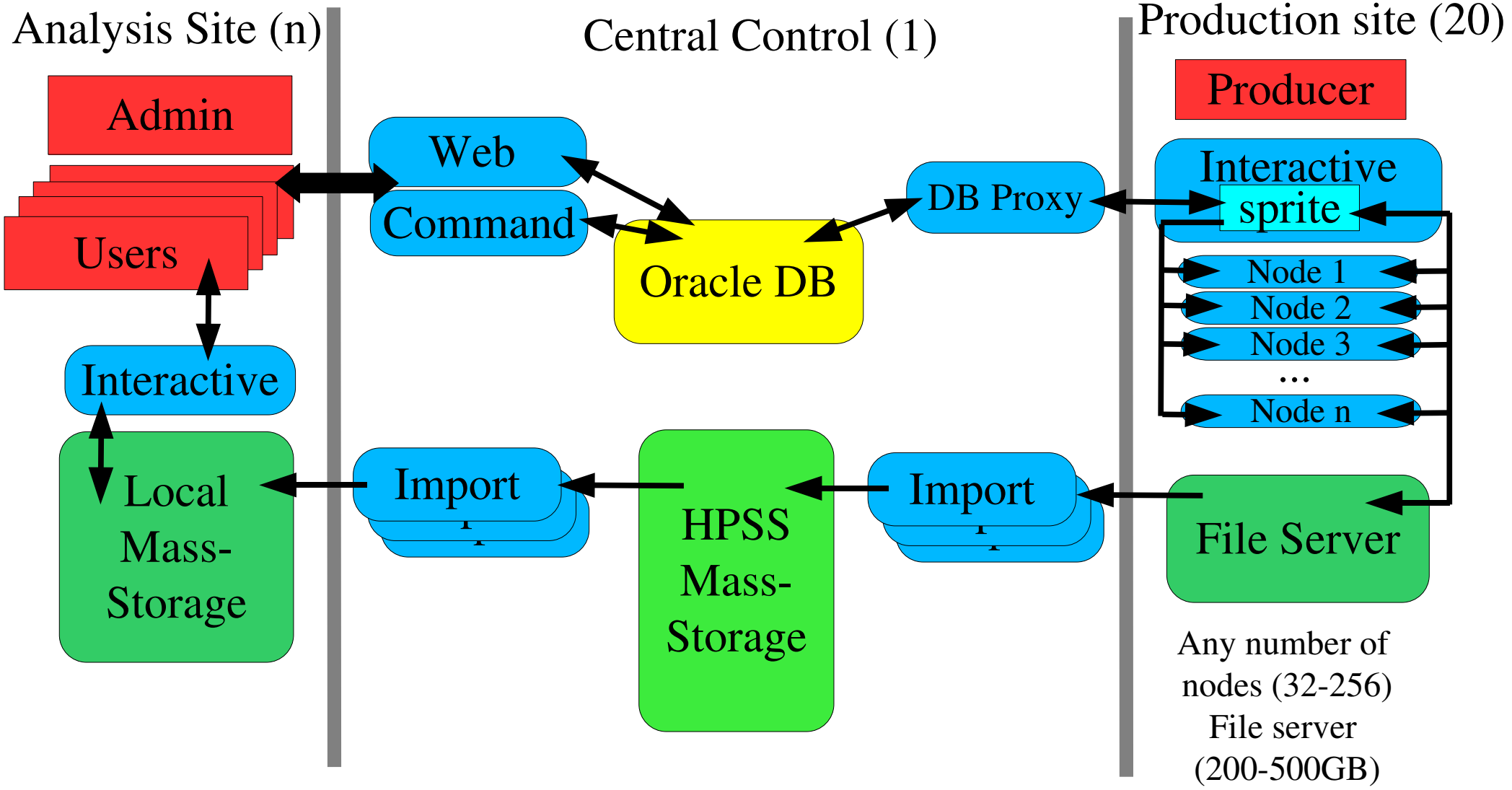


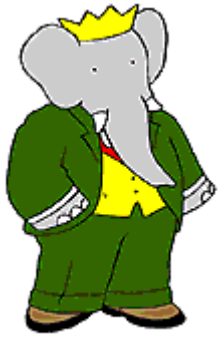
Comparison of production

- When using database:
  - Get setup and submit job. (spbuild, spsub).
  - One database, all events together.
  - Databases no longer touched, export them (MocaExpresso).
  - Check databases and attach to analysis database for use.
- When producing root files:
  - Get setup and submit job. (spbuild, spsub).
  - Many small files produced -- merge together into collections of similar events (spmerge).
  - Export complete files for use (spexport).
  - Check import -- mark data as finished (spimportcheck).



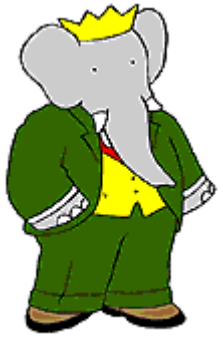
Cartoon of production




TM & © Nelvana

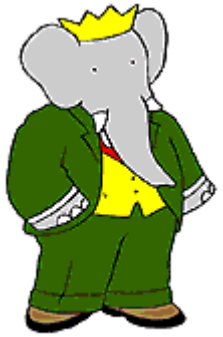
Sprite

- To reduce effort, a run control daemon was created: sprite.
- Each job has small, trusted, status file: checked at ~1000 jobs per sec.
- Checks status files, and decided what to do, and does it.
  - If low on jobs, build more getting config from central database.
  - If jobs needed fixing from failures, fix them.
  - Submit jobs to keep some constant level in the queue.
  - Merge output if there is enough.
  - Export completed collection to SLAC, and clean up.
- Includes lots (-LOTS-) of error checking, pretty much if anything goes wrong, it knows what to do, after 4 years in use now.
- Full set of site configurations.



Simu. Production (SP) cycles

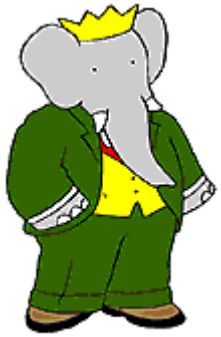
- Production in BaBar divided into cycles, and numbered.
- SP5 was last CM1 production cycle, was converted in CM2.
- CM2 production cycles are now SP6 and SP8 (SP7 was not done due to production schedules changes from accident).
- Production cycles are based on major software releases, and need to match data run cycles (also just numbered 1-5). SP5 used for analysis of Run 1, 2 and 3; SP6 used for analysis of Run 4; and SP8 used for analysis of Run 1-5.
- Luminosity of each data run cycle determines size of requests for simulation cycle.



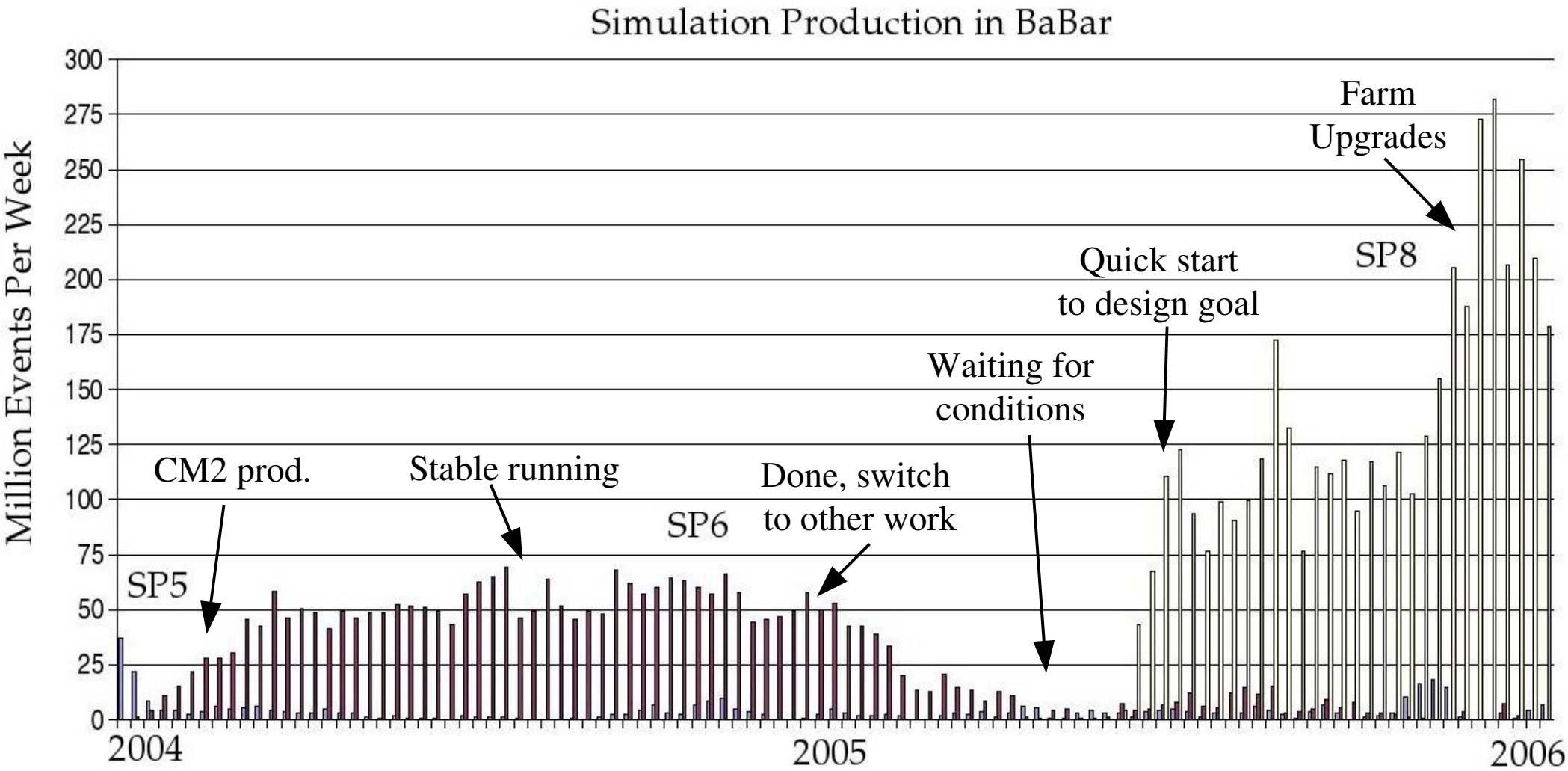
Production Numbers

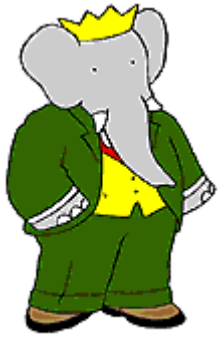
Cycle	Jobs (M)	Events (B)	CPU time (y)	Data (TB)	Files (k)
SP5	1.5	2.0	415	28.6	81
SP6	1.6	2.9	987	44.8	65
SP8	2.4	4.5	819	62.2	97

- A couple nice things to notice in these numbers
  - Numbers are fairly large, but not so bad if you have ~2000 cpus, and 100TB storage.
  - From SP5 to SP6 the number of files greatly reduced to save more data, better for storage and distribution.
  - From SP6 to SP8 amount of CPU time reduced, even though significantly more events produced. Due to farm upgrades to better cpus (and about 20% due to software changes).



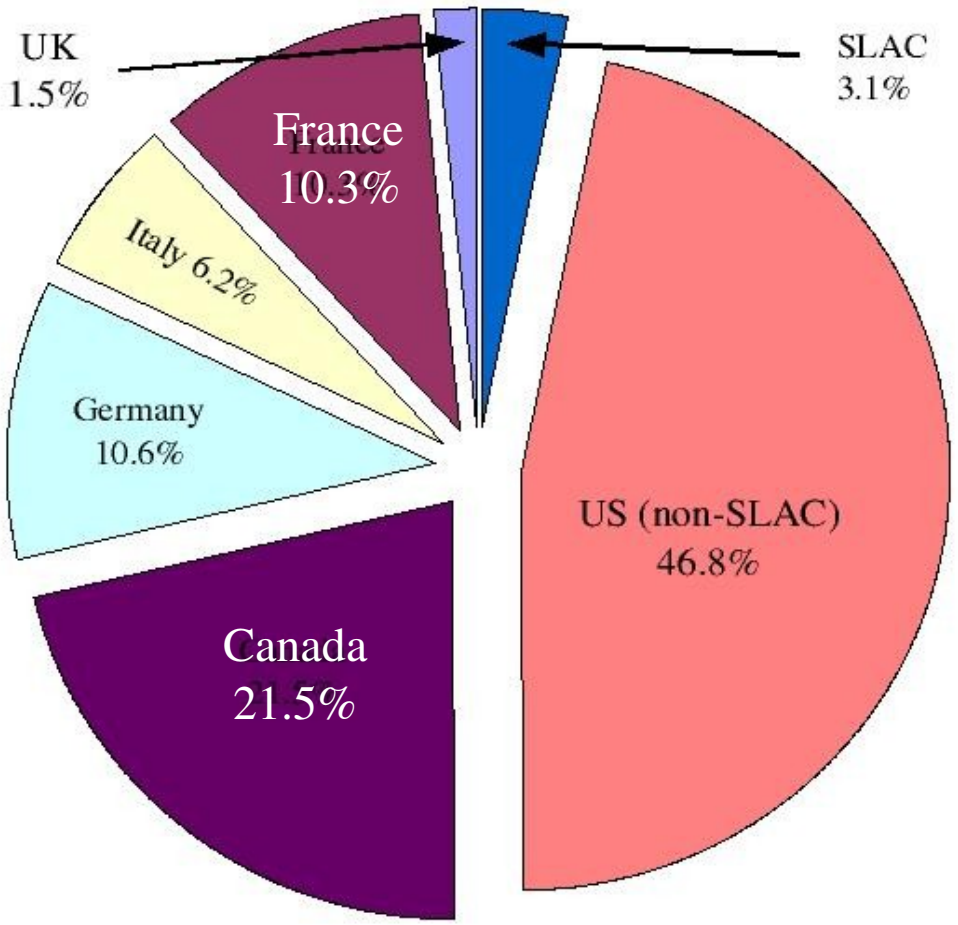
Production by week





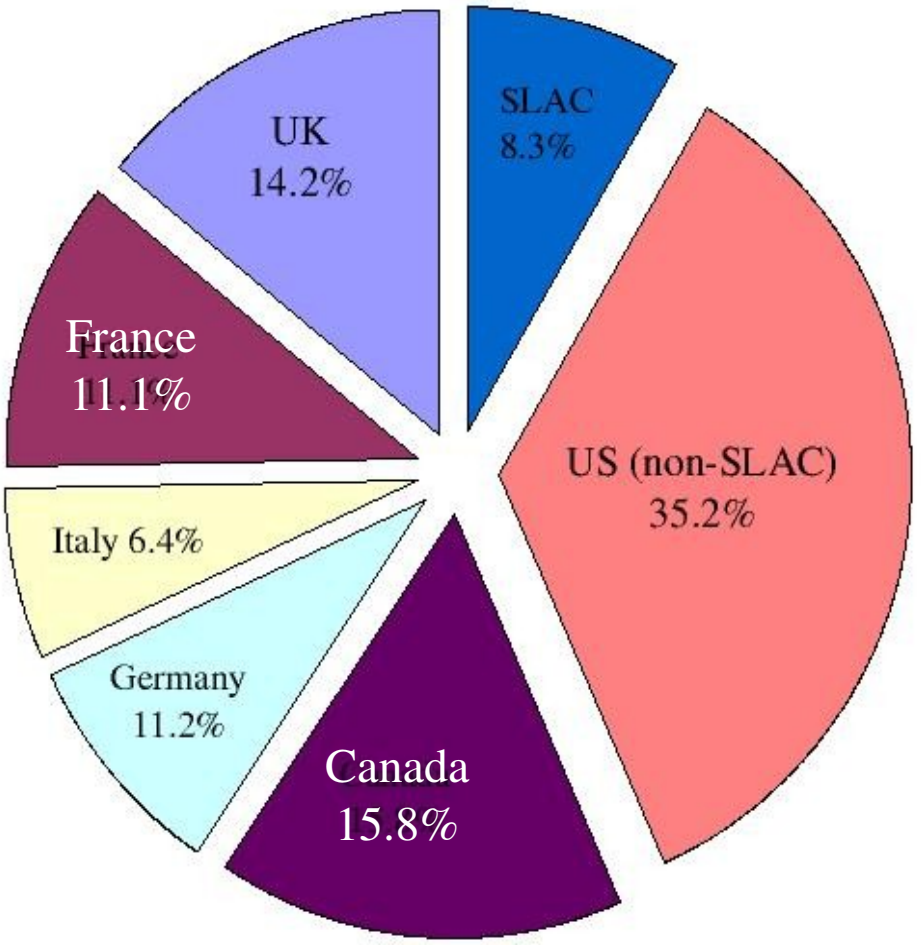
Production by country

SP6

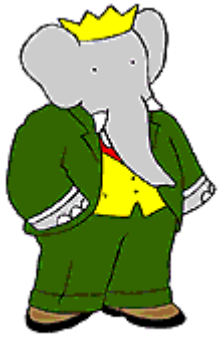


2.9 billion events

SP8

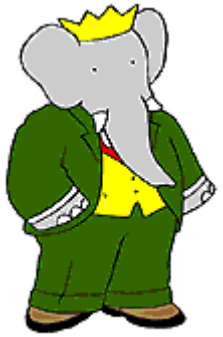


4.5 billion events



Results of changes

- Removal of database increased stability
  - Do to database, failure rate was 4-6%
  - After removal, almost no failures due to data storage, now ~1% due to infrastructure, about 1/3% due to software (now fixed and is less than 0.1%).
- Less effort for production
  - Site managers do not have to be database admins, less to do.
  - Import/Export of production now just files, less work to get into analysis. (Latency for use – from 2 weeks to now ~24 hours.)
  - Simple to now tell which data is involved with which files.
- Has been just “a good thing”



Production status

- Moving to CM2 in BaBar increases efficiency of production.
- In latest cycles SP6 and SP8 were able to scale to larger rates, but with less effort.
- SP6 was designed to meet physics requests in 9 months of production, and it did just that, even as the first production using CM2 data.
- SP8 was designed to meet physics requests in 9 months, but producing many times what SP6 did. Because of compute farm upgrades and ability to scale with good efficiency requests should be met in 6 months.
- Life without a database for data is much better.