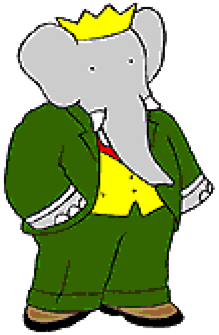
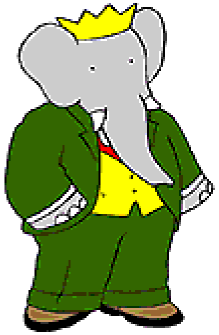


Data management in BaBar

Douglas A. Smith
Stanford Linear Accelerator Center
Tim Adye
Rutherford Appleton Laboratory

for the BaBar computing group

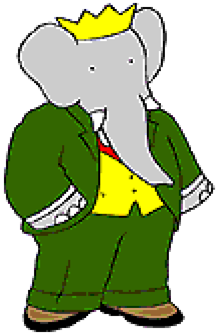
CHEP 2007
Sep 3rd, 2007, Victoria, BC, Canada



TM & © Nelvana

BaBar Event store

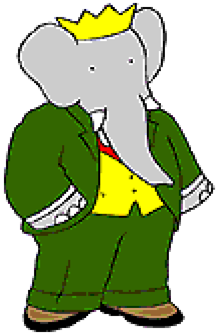
- The event store contains **all** the data in BaBar.
  - reconstructed event data, simulation data, skimming streams, conditions data, random trigger data, user generated data, ...
- With various processing and re-processings of these.
- Stored as a set of files on storage devices, the events are kept in “collections” (of 1000's to 100,000's).
- Unique accessor of data is the collection name (you provide a collection name to the framework for access to that data).



TM & © Nelvana

BaBar Bookkeeping

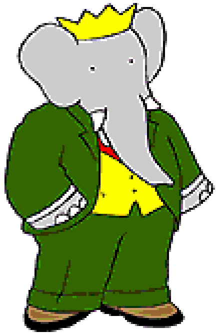
- The meta-data on collections in the event store is the BaBar Bookkeeping System.
- In use in BaBar for years now, providing fast access to data.
- A set of utilities and libraries to access the meta data, which is stored in a relational database.
- Implementation details not important for this talk, for details see CHEP2004 and CHEP2006 talks for more info.
- This talk is will be more abstract about data management, and how it is done in our working system.



TM & © Nelvana

Use of data

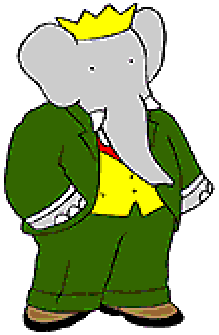
- No one wants **all** of the BaBar data.
- Each user in BaBar sees in the event store through his own eyes, and only thinks of, or desires, "his" data.
- There are ~2M collections of data in the event store, a user might only be interested in a few 100 to a few thousand collections of data.
- Collections of events, are collected together into sets of similar data for use, in a “dataset”.



TM & © Nelvana

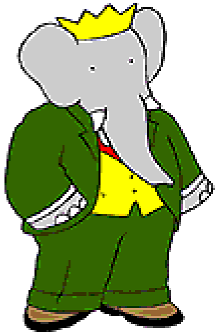
Datasets

- A dataset is a list of collections.
- Relation between datasets and collections is *n-to-m*.
- A dataset should be able to store a list of any random data.
- Users access the system with a dataset name to list the collections that are "her" data.
- Many types of data, so many datasets, along with sub-sets of data (run cycle, month, on or off peak...).
- Keeping separate data separate results in a need for ~140,000 datasets. (Even if there are only a couple hundred users at any given time.)



Data in times of change

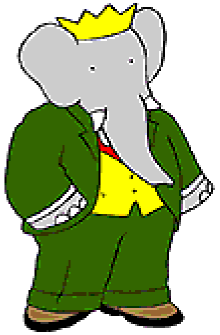
- Challenges arise with data changes -- data is **always** changing.
- People think of "their" data as a static set, but the first complaint is "why isn't this data up-to-date". Data always has to change, and quickly -- and data always has to stay the same.
- All lists and changes are stored in a database, no matter how many changes.
- As data changes, can't store a new list for each dataset, too much information.



TM & © Nelvana

Storage of changes

- As is usually done -- you don't record the list, you record the changes.
- The database has a table of collection names, a table of dataset names, and the contents of a dataset are stored as an association between these in a third table.
- The association is stored by date, and can be an "addition" or "removal".
- A dataset list is created by selecting links in time order, applying the additions and removal as you go, until you get to a certain date.



TM & © Nelvana

Cartoon of storage

Collection table

1. reco-coll01
2. reco-coll02
3. reco-coll03
4. simu-coll01
5. simu-coll02
6. reco-coll04
7. reco-coll05
8. simu-coll03
9. reco-coll06
10. reco-coll07

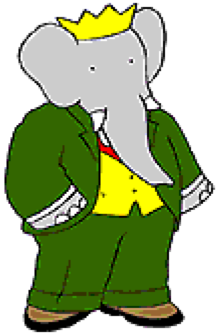
Links Table*

1. Jan 1, add 1 to 3
2. Jan 3, add 2 to 3
3. Jan 5, add 3 to 3
4. Jan 11, add 6 to 3
5. Jan 17, add 7 to 3
6. Jan 18, rm 3 from 3
7. Jan 19, add 9 to 3
8. Jan 23, add 10 to 3

Dataset name table

1. Reco-Stream01-R20
2. Simu-Mode01-Stream01-R20
3. Reco-Stream02-R20
4. Reco-Stream03-R20
5. Simu-Mode02-Stream02-R20

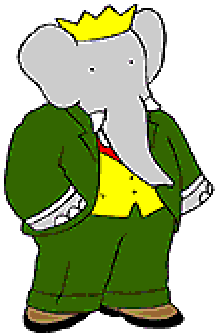
* only listing links for table 3, to simplify example.



TM & © Nelvana

Update of changes

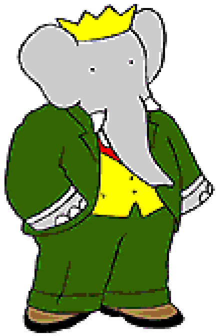
- Initially defined a dataset as a set of selection rules on the collection list (type of data, run cycle, decay mode...)
- A collection list based on these rules was selected, and compared to the current dataset, and changes were applied.
- This required about 10 sec. of cpu and server time to update.
- At the time there were ~25,000 datasets, resulting in about 70 hours of cpu time to update.
- Parallelized : some datasets kept up to date daily, and others weekly.
- It was not acceptable to the collaboration, resulting in constant uncertainty in the state of the data. Something had to be done to be faster than that.



TM & © Nelvana

Keeping up with change

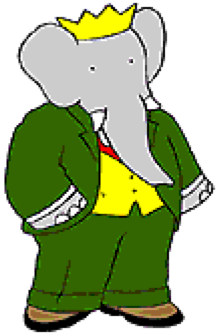
- Reversed the problem -- use selection rules, and then for each collection generate a list of datasets.
- Only use collections changed in some recent time, then apply changes to effected datasets.
- If you keep the "recent time" small, and update often, the list of effected datasets and collections can be kept up to date within 1-2 minutes.
- Datasets are kept up to date within 15 mins, and users don't notice differences between data and datasets.



TM & © Nelvana

Fixing data, no more change

- Can have datasets "tags", this is taken from the CVS model.
- The "tag" is a date, all changes to a dataset are selected until that date.
- Stored in a table of tag names, people prefer this instead of a dataset and tag date.
- These tags are usually used in a way to present a similar set of data for all analyses -- like for a conference -- thousands of datasets will be tagged to provide the same luminosity.

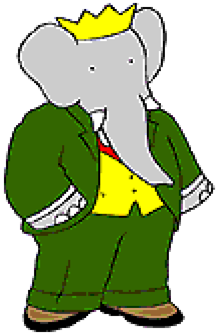


Adding data to an analysis

- If an analysis was done for a tagged dataset, you might want to only add data since that tag.
- System provides for this, you can create an update of data since a certain date, or for a set of data in a dataset between two dates.
- The system will select all changes to a dataset between two dates, providing a list of update data for an analysis.
- This is mostly data added to the dataset, but can also be removals due to quality changes. Removals will be reported to users, so they can remove this data from their current analysis.

Uses of datasets in BaBar

- Use for analysis
  - To keep data separate for each analysis. The analysis user will have a list of datasets, usually a few 10's of datasets. These will be used as input to analysis jobs.
- Use for distribution
  - Analysis is distributed to remote and certain sites will import certain data. Also on demand a user can request a certain dataset for import to anywhere. (local academic sites, laptops).
- Use for production
  - Datasets of data to be used in production at sites. To be used as input to production jobs, which produce more data for analysis.



TM & © Nelvana

Conclusions

- Bookkeeping system works well in keeping track of millions of collections of data, and keeping the data separate for 100's to thousands of different uses.
- Allows for random lists of any type of data.
- Data always changes, and any system needs to be able to keep up with changes, and changes can be very rapid.
- Any of the datasets can be "tagged" to provide a fixed set of data, even while the dataset continues to change.