BaBar Bookkeeping – a distributed meta-data catalog.

D. A. Smith, T. Adye, D. Bukin, A. Ceseracciu, G. Dubois-Felsmann, A. Forti, D. Hutchcroft, P. Jackson, D. Kovalskyi, W. Roethel

for the BaBar computing group

CHEP 2004 - [338] - Sep 30, 2004

BaBar Computing Model 2

- Beginning of last year, BaBar changed computing model, from a database based event store to an event store based on root files.
- Along with this change, people wanted a new meta-data catalog, which would hopefully be a fast way of listing the data needed for analysis.
- It should scale to handle all data for the life of BaBar, and be flexible to provide what people need for each analysis.
- Also it needed to be distributed so people can access data at any computing center in BaBar.
- Requirements resulted in the BaBar Bookkeeping project -- a set of command line tools and libraries, which interact with a relational database to store the information.

Talk by Pete Elmer – id 502, 11:30 Mon.

Collections

- The event store is made up on unique units of data, we call “collections” (they are collections of events).
- Each collection has a unique name.
- This is the heart of the bookkeeping, a list of all the collections which exist in the BaBar event store.
- Attributes on collections are also recorded, such as QA status (good or bad), data type (real or simulation), run cycle, etc...
- Examples:

- Processing --> /store/PR/R14/AllEvents/0004/96/14.4.4e/AllEvents_00049689_14.4.4eV01
- Simulation --> /store/SP/R14/001237/200407/14.4.3a/SP_001237_016169
- Skimming --> /store/PRskims/R14/14.4.3d/AllEvents/13/AllEvents_1317

Collection files

- Collections are organized to best simplify the event store (similar events are merged together into a collection when possible), and provide for the easiest archiving and data distribution (file size should be close to 2GB as possible).
- The implementation of a collection is a set of root files. Each event in BaBar has different components, and these components can exist in separate files.
- The bookkeeping will keep track of the files which exist as part of a collection in the event store.
- Information on Logical File Names (LFN) is stored, independent of BaBar sites or servers.

Talk by - Matthias Steinke (Pete Elmer), id 172, 17:10 Wed.

Runs

- The experiment is performed with a detector, and the data from the detector is divided into runs.
- Runs are not a unit of data in the event store. Each run needs to be processed, and there can be several versions of processing.
- Each job could be over one or many runs, making the relation between runs and collections *n to m*.
- But the run is what we call a "Non-Overlapping Unit" of management, so we have to record this and make sure there are not two versions of the same run in any analysis.
- Example:
  - /store/PRskims/R14/14.4.3d/AllEvents/13/AllEvents_1317 --> contains 23 runs.
  - Run 49670 --> part of 127 collections.

Datasets

- Knowing all the collections doesn't give the people what they want, different analysis need different parts of the event store.
- Bookkeeping provides what people need in the form of Datasets.
- These are just lists of collections, based on similar attributes. (data or simulation, run cycle, on peak or off peak, so on...)
- Each dataset has a unique name, and provides fast access (usually a few secs.) of what you need; and ease of use, only need to know the dataset name.

Example (a simple one):

```
prompt> BbkUser --dataset SP-uds-AllEventsSkim-Run4-R14 collection
COLLECTION
/store/SPskims/R14/14.4.3d/AllEvents/00/000998/200310/AllEvents_000998_1539
/store/SPskims/R14/14.4.3d/AllEvents/00/000998/200309/AllEvents_000998_1540
2 rows returned
```

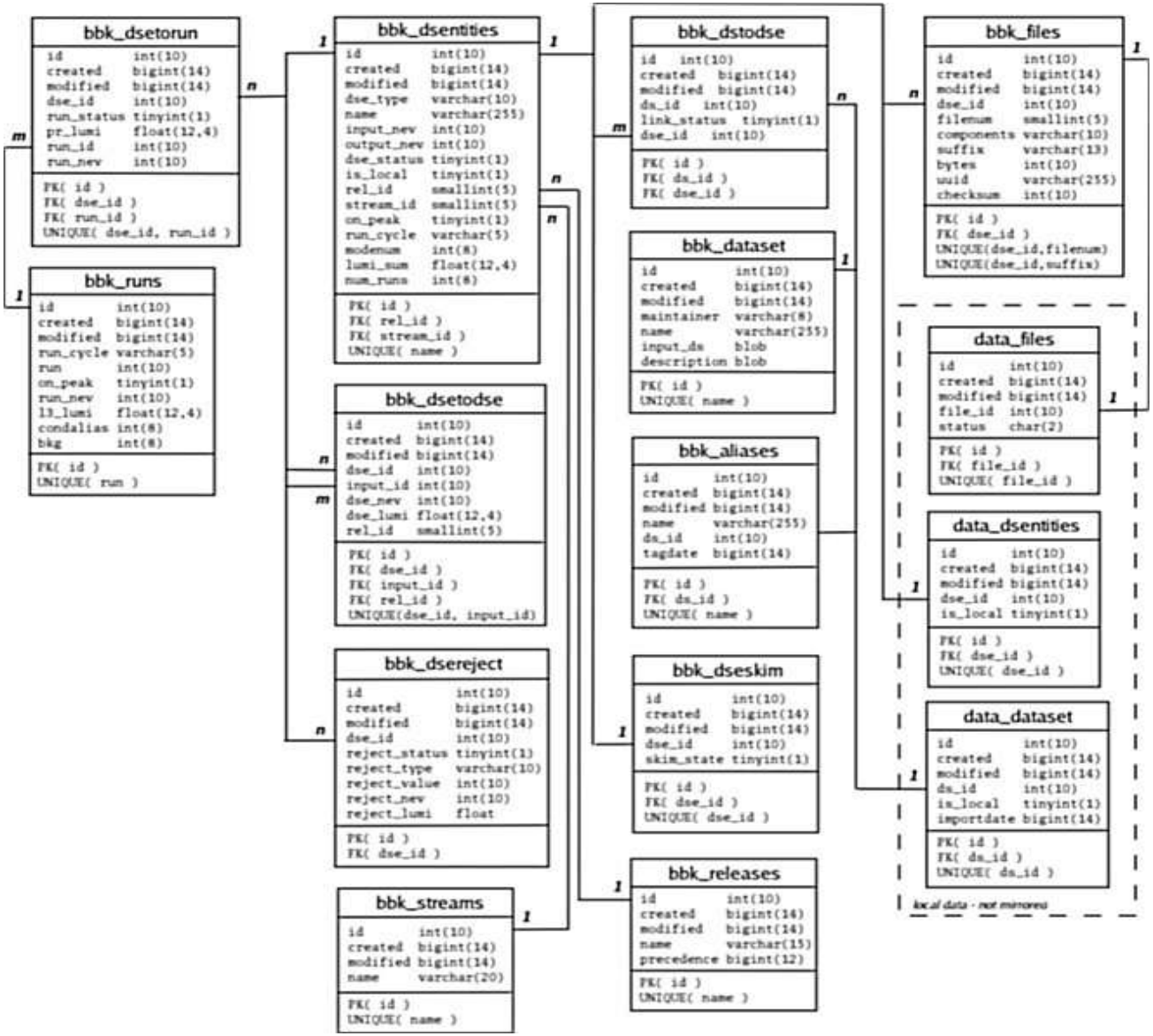
Dataset evolution and tags

- The event store changes and evolves, almost hourly and the datasets need to stay up to date.
- Analysis also needs stability, to know exactly what was used when, and to ignore further changes to the dataset.
- Bookkeeping provides dataset tags, in a model similar to CVS. These tags have unique names, which will be stable and never change, while the datasets continue to get updated with further production.

Examples:

```
AllEventsSkim-Run4-OnPeak-R14-GreenCircle     -- 44 collections
AllEventsSkim-Run4-OnPeak-R14-BlueSquare      -- 66 collections
AllEventsSkim-Run4-OnPeak-R14-BlackDiamond    -- 76 collections
AllEventsSkim-Run4-OnPeak-R14                 -- 80 collections
```

Database schema



- Each box is a table in the database.
- The schema is more than just 3 tables (for collections, runs, and datasets), but still fairly simple.

SQL Selection API

- User will need to interact with the data in the database, but the developers can't really (or don't want to) create all SQL select statements that a user might need.
- All columns in database tables are given an alias, and user can select these aliases given conditions on other aliases, and the SQL statement will be automatically generated.
- This is a general purpose API created for this system, but should work for any set of relational database tables.

SQL Example

Example of SQL API use – select runs based on a collection :

```
prompt> BbkUser --collection /store/PRskims/R14/14.4.3d/XiMinus/15/XiMinus_1550 run
RUN
50488
50489
<...more runs...>
50538
48 rows returned
```

Actual SQL created:

```
SELECT   bbkr14.bbk_runs.run
   FROM bbkr14.bbk_dsentities,
        bbkr14.bbk_runs,
        bbkr14.bbk_dsetorun
  WHERE bbkr14.bbk_dsetorun.dse_id=bbkr14.bbk_dsentities.id
    AND bbkr14.bbk_runs.id=bbkr14.bbk_dsetorun.run_id
    AND bbkr14.bbk_dsentities.name = '/store/PRskims/R14/14.4.3d/XiMinus/15/XiMinus_1550';
```

Distribution Features

- BaBar is large collaboration, with analysis going on at many sites. There is a need for bookkeeping at more than one site.
- Also each remote site usually has only part of the event store, bookkeeping will keep track of which datasets and collections are local.
- The entire database can be mirrored to any site on demand, and changes to database can be updated to keep things in sync.
- Network access to each database is also granted to all BaBar members, so remote sites can access any of the mirrored databases.
- In the end any BaBar member can access the meta-data from anywhere.

TM & © Nelvana

Connection distribution

- To be able to distribute the database information a database key distribution system was created.
- This will distribute the definitions for the database connection along with the connection keys on demand, with very little user interaction.
- Authentication based on unix account at SLAC and use of ssh.
- Scalable for multiple sites, servers at each site, and users in each server, each can have different definitions and permissions.
- Users with the same tool can access into on Oracle at SLAC, and MySQL at RAL, without needing to know how the connection is made.

Database mirroring

- System needed to support Oracle and MySQL, since these were the relational database systems already in use.
- To distribute access load and bookkeeping of local data, it was needed to mirror database records between different database servers.
- Mirroring happens on request, and pulls from SLAC, the central production database. Inserts and updates only happen to the SLAC database.

Distribution of data

- Along with distributing the meta-data, the bookkeeping includes data import and export tools, to distribute the data.
- The import/export tools are driven by the bookkeeping database.
- Data is distributed based on the defined datasets and collection components (i.e. micro or mini), so people can choose which part of the event store they need.
- This scales from large sites which want most of the event store, to a laptop with only a few collections.

Current Status

- The system was used to provide data access for the latest run cycle of the BaBar detector, and all data converted to the new event store.
- Beta tested last fall and winter, and fully used in production since Feb. of this year.
- Now contains ~1M runs, 290k files, 184k collections and 17k datasets, and the total size of the database is about 4GB. Compared to the total event store which is 161TB.
- Most people are happy with the new system (we believe).

Task Management

- A large part of the system is providing task management.
- A task can be defined on a dataset, this task is a set of jobs which need to be run over the collections in the dataset.
- The system includes job setup, submission and management, and has been used this year for production skim processing.
- For more details there is a poster which people should see.

Poster by W. Roethel, D. Smith – id 350, 10:00 Wed.

Conclusions

- BaBar has successfully moved to a Root I/O event store, and the Bookkeeping project has supplied the meta-data catalog.
- The system is distributed to BaBar sites on demand, and will naturally scale to meet computing needs.
- Using datasets for people's needs, the system can scale nicely to provide what people want quickly.
- With built in data distribution tools provides an easy to use solution for data access from large computing centers down to laptops.