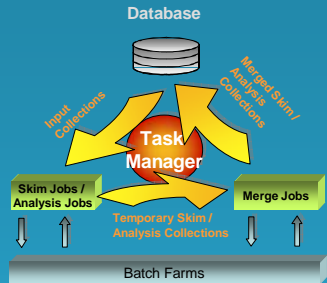


# BABAR Analysis Task Manager

W.Roethel<sup>1</sup>, D. Smith<sup>2</sup>, T. Adye<sup>3</sup>, D. Bukin<sup>4</sup>, A. Ceseracciu<sup>5</sup>, G. Dubois-Felsmann<sup>6</sup>, A. Forti<sup>7</sup>, D. Hutchcroft<sup>8</sup>, P.S.Jackson<sup>9</sup>, D. Kovalskiy<sup>10</sup>

## For the BABAR Computing Group

<sup>1</sup> University of California, Irvine; <sup>2</sup> Stanford Linear Accelerator Center; <sup>3</sup> Rutherford Appleton Laboratory; <sup>4</sup> Budker Institute of Nuclear Physics; <sup>5</sup> Università di Padova; <sup>6</sup> California Institute of Technology; <sup>7</sup> University of Manchester; <sup>8</sup> University of Liverpool; <sup>9</sup> University of London, Royal Holloway and Bedford New College; <sup>10</sup> University of Maryland



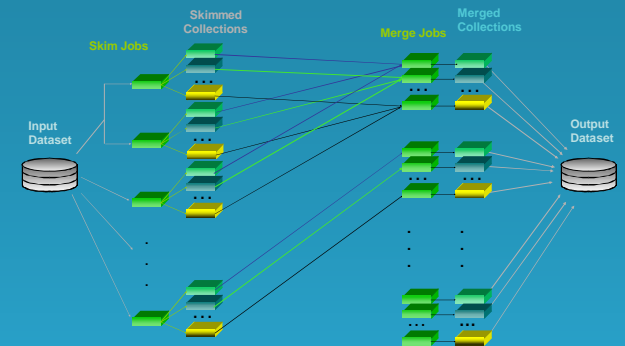
### BABAR Analysis / Skim Production Workflow

Basis for any physics analysis of BABAR data, or the central production of skims, is a list of data collections provided by the bookkeeping database. Given this list, users create a set of jobs (O(1000)) using the corresponding data as input, run these jobs on the batch farms and collect the output, which can be new data collections, nuples or text files. In the case of data collections, several of these are merged together into sets of larger sized collections, which then can be imported into the main event store and the bookkeeping database.

The Task Manager provides a framework for data analysis or data production by combining the steps listed above into one environment. Using a given input dataset the Task Manager creates the set of jobs, provides an interface for running these jobs on the batch farms (currently supporting LSF and PBS), manages the subsequent merges over the output of the analysis or skim production jobs, and allows the imports of these merged collections into the BABAR event store and bookkeeping.

### Skim Production

- The Input Dataset contains ~20,000 collections with an average size of 120,000 events.
- Input collections are divided into several jobs, each job processing ~25,000 events.
- Every skim job produces ~120 output collections of moderate size (50MB), i.e. one output collection for every skim.
- The skimmed output of ~1000 skim jobs are merged together into one single collection. (Fewer, large collection are easier to manage by users and show substantial performance advantages in the event store).



### The BABAR Bookkeeping Database

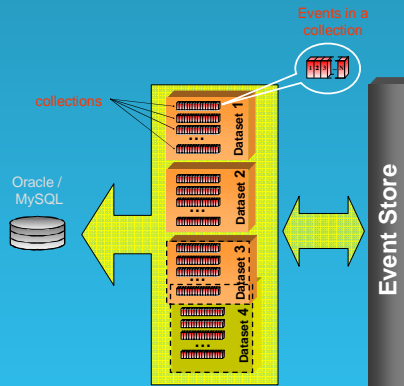
BABAR data is stored in collections (of events) with, in the context of BABAR's Computing Model 2, every collection corresponding to a set of root-based files in the event store. The BABAR Bookkeeping organizes the collections in datasets, which are simply lists of collections with similar, well defined, attributes, typically covering all data relevant for physics analysis. Users looking for data to run analyses simply have to select the particular datasets in order to retrieve information.

The BABAR Bookkeeping offers further features for datasets:

- Datasets evolve over time, i.e. new collections can be added or can replace other collections. This allows users to get the most up-to-date set of BABAR data.
- Datasets can be tagged and named. Tagged datasets are 'snapshots' of a given dataset at a given time and do not change over the lifetime of the BABAR bookkeeping.
- By providing a date it is possible to retrieve the status of a dataset at any given point in time.

The BABAR bookkeeping also adds features for distributed computing:

- Tools for mirroring the complete database, or selected datasets only.
- Integrated data distribution for easy imports of BABAR data to local sites.
- Integrated networking with a unique authentication method which provides passwordless access to any mirror of the BABAR bookkeeping database.

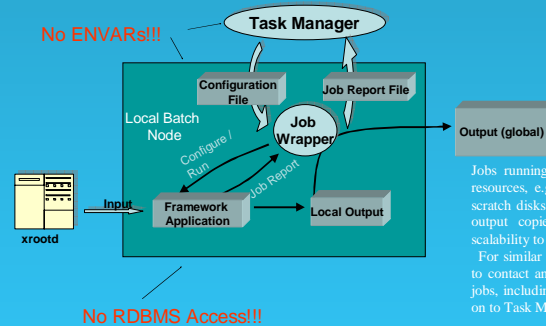


### Running on the Batch Nodes

The offline applications running on the batch farms are managed by a job wrapper and configured over a configuration file which contains all information on the run environment and the processing itself. This avoids the use of any environmental variables to pass information to the analysis / skim production application.

Jobs running on the batch farms heavily rely on the use of local resources, e.g. all output is written to temporary locations on local scratch disks and only upon successful completion of the job, is the output copied to a globally accessible location. This provides scalability to the requirement of over a thousand concurrent jobs.

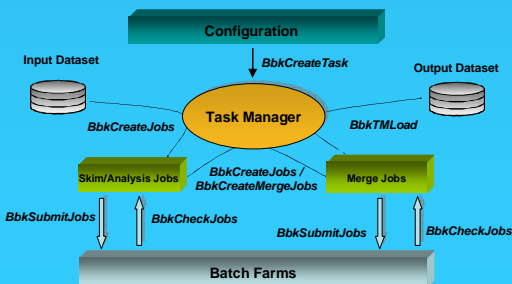
For similar reasons jobs running in the batch farms are not allowed to contact any relational database. Information on the processing of jobs, including all information to be stored in the database, is passed on to Task Manager framework over job report files.



### User Interface

The Task Manager is organized in "Tasks", with a task being defined as a set of jobs that share a common run environment, e.g. the same offline analysis application. The configuration of the Task contains all information on how the analysis/production job is to be run, i.e. the executable, the location of the output, size of analysis/skim production jobs and the subsequent merges of their output, etc. The Task is created and configured by running BbkCreateTask. All remaining production efforts can then be managed with just a few commands:

- BbkCreateJobs / BbkCreateMergeJobs – creates the analysis / skim production jobs or the jobs to merge single output collection to run on the batch farms.
- BbkSubmitJobs / BbkCheckJobs – submits jobs to the batch queue and check the success of jobs that have completed.
- BbkTMLoad – imports the merged collections produced by the analysis / skim production into the BABAR bookkeeping database.



### Distributed Production

The Task Manager combined with the dataset bookkeeping allows for simple and straight forward distribution of skim production efforts:

- The input dataset for overall production is distributed over several datasets, with one dataset for each production site.
- After a database mirror the data distribution tools, as part of the BABAR bookkeeping, identify collections that are listed in the production dataset but not present in the local event store inventory of the production site and import these to the production site.
- The Task Manager manages the skim production on the production site.
- Merged skim collections are imported into the SLAC bookkeeping database and files associated with skimmed collections are transferred to SLAC.

This distributed production system has been used successfully in recent skim production efforts at SLAC (US), GridKa (Germany) and Padova (Italy).

