



TM & © Nelvana

# Babar Task Manager II

---

Douglas Smith

*SLAC National Accelerator Laboratory*

Tim Adye, Chris Brew

*Rutherford Appleton Laboratory*

Gregory Dubois-Felsmann, Bertand Echenard

*California Institute of Technology*

Wilhelm J. Roethel

*University of California, Irvine*

Andreas Petzold

*Technische Universitat Dortmund*

for the Babar computing group

Mon, Mar 23, 2009



TM & © Nelvana

# History

---

- Development of a system to define tasks against datasets, manage jobs, and produce data into new datasets was begun in 2003.
- First version in use 2004 to start of 2007.
- Problems with initial design of task manager.
- A major effort of redesign in 2005-2006 to create a new task manager and allow use of Grid.
- Production tests for second version started end of 2006 – Task Manager II.
- Development both versions done by Will Rothel.



TM & © Nelvana

# Use in Skim Production

---

- Ability to define tasks applied to datasets to produce data into new datasets natural for Skim Production.
- In Babar, data is reconstructed and simulated, but this isn't of use for all analysis – data needs “Skimming”.
- Events are selected from the data, and often new data is added to the events. The selection definition is called a “Stream”.
- A full skimming is called a “cycle”, this is a set of streams derived from the data. Skim cycles in Babar contain any number of streams, from 10-200.
- New data is produced in each cycle from each stream.

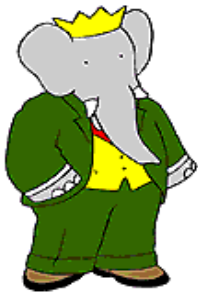


TM & © Nelvana

# Use of Task Manager II

---

- Initial version used for skimming and was causing problems, not able to handle scaling to required number of streams produced (~200 at the time).
- A serious problem in early 2007, and use of Task Manager II was put into production that spring.
- This has been used for all Babar skim production since that time.
- Ten skim cycles have been processed using the system at this time.

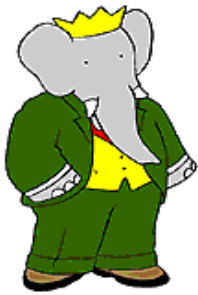


TM & © Nelvana

# What it does

---

- Creates a task, which defines a generic job on data from a dataset.
- Creates specific jobs based on the task definition, and the amount of data in the dataset.
- Produced data is merged into a set of output collections for use.
- Collections are transferred to SLAC for archiving and distribution, and info. is put into bookkeeping database.
- A management daemon keeps this running smoothly.

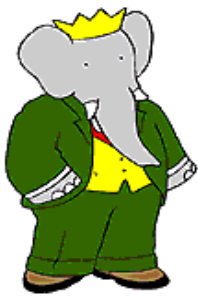


TM & © Nelvana

# System design

---

- The system was developed a set of object oriented perl utilities and modules.
- The control of all task information is in a relational database back end.
- There is a common db-interface to allow use of Oracle or MySql as the back end.
- Database contains all task and job state information at the production site, each independently controlled.



TM & © Nelvana

# Config system

---

- All definitions are set in configs, no code changes needed for separate tasks with different jobs.
- Number of configs is lengthy at this point, i.e. release dir., shell script wrapper templates, batch system commands, data storage systems and so on.
- Multiple configs can be sourced, so small task configs can source large common configs.
- For production common configs are sourced by site configs, which are sourced by the task configs at each site.



TM & © Nelvana

# Creation of skim jobs

---

- A skim “super” is defined for each input collection.
- Jobs of appropriate size are created, of so many events each. Any number of skim jobs for each skim super, usually 5-20.
- A set of configs and a wrapper shell script in a working dir. are created, and jobs submitted.
- Data goes to local storage, separate from the job dir.
- Status is monitored, checked, and updated. Once done the output data is checked as good.



TM & © Nelvana

# Cartoon of skim jobs

Input Dataset01:

Coll01

Skim Super 01

Job01

Job02

...

Job08

Output Data:

Out01-001

Out01-002

...

Out01-056

Out02-001

Out02-002

...

Out02-056

Out08-001

Out08-002

...

Out08-056

Skim Super 02

Job09

Job10

...

Job21

Out09-001

Out09-002

...

Out09-056

Out10-001

Out10-002

...

Out10-056

Out21-001

Out21-002

...

Out21-056

... and so on...

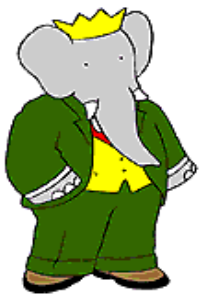


TM & © Nelvana

# Creation of merge jobs

---

- Skim job output can be small, too many small files to be useful for storage and analysis.
- As good skim output is created merge “supers” are created.
- All output from a skim super is good, then a set of skim supers are defined as input for a merge super.
- With many skim outputs, a separate merge job created for each output stream of the skim jobs.
- Again jobs submitted, status monitored, output is checked to be good.

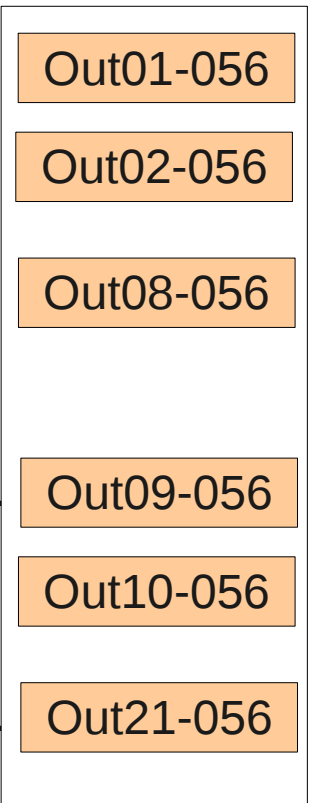
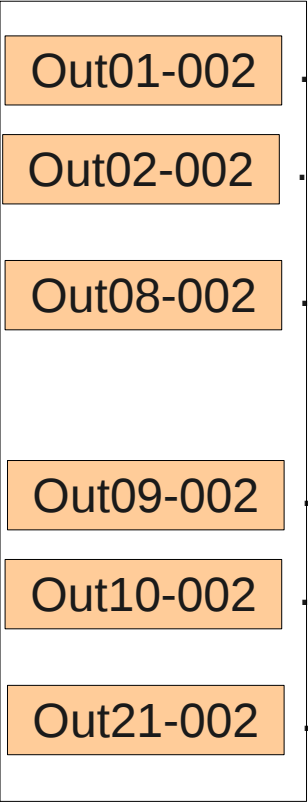
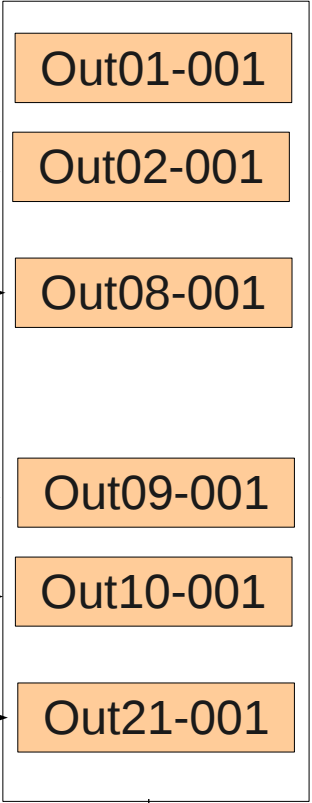
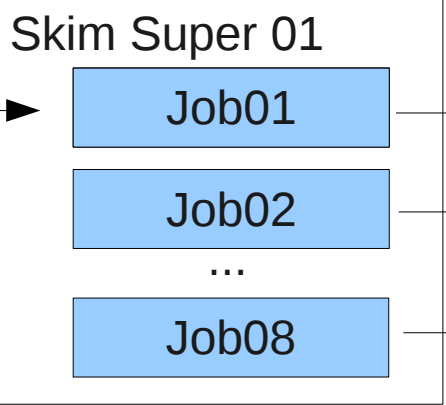


# Cartoon of merge jobs

TM & © Nelvana

Input Data:

Coll01



... and so on...

Merge001

Merge002

Merge056



TM & © Nelvana

# Handling output

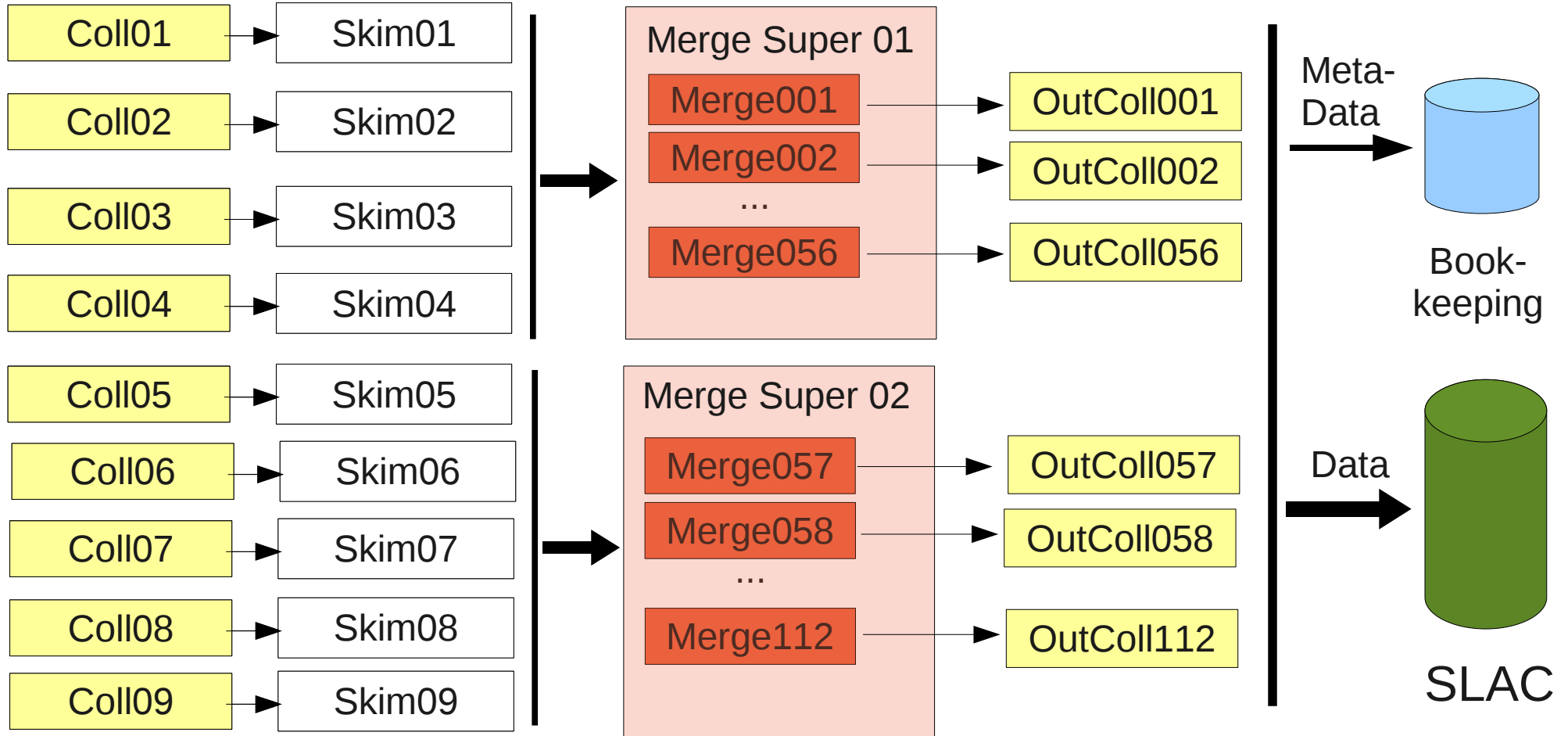
---

- Merged output is now ready for archiving and use.
- As merge jobs are found to be good, output is transferred to SLAC for archiving.
- Info on jobs is put into the bookkeeping system, with all info about data and files produced.
- Once all output from the merge super is done and ready, then output is called good and is ready for use.



TM & © Nelvana

# Merge Output



Mon, Mar 23, 2009

13





TM & © Nelvana

# Future

---

- System fairly stable at this time, partly due to success, partly due to lack of man power to continue dev.
- Currently in use at SLAC and GridKa, soon will be setup at IN2P3.
- As new data is produced in Babar, through to the end of this year. At least a few more skim cycles in the plans, and the system will be used on these.