BNL-95311-2011-CP

PX. No. 459

Analyzing Ever Growing Datasets in PHENIX

Christopher Pinkenburg[1], et al.

[1]Brookhaven National Laboratory, Upton, NY 11973 USA

Conference CHEP 2010, Computing in High Energy Physics
Located in Taipei, Taiwan

October 18-22, 2010

Physics Department

Brookhaven National Laboratory

U.S. Department of Energy
DOE Office of Science

DISCLAIMER

Analyzing Ever Growing Datasets in PHENIX

Christopher Pinkenburg for the PHENIX collaboration

Physics Department, Brookhaven National Laboratory

E-mail: pinkenburg@bnl.gov

Abstract. After 10 years of running, the PHENIX experiment has by now accumulated more than 700 TB of reconstructed data which are directly used for analysis. Analyzing these amounts of data efficiently requires a coordinated approach. Beginning in 2005 we started to develop a system for the RHIC Atlas Computing Facility (RACF) which allows the efficient analysis of these large data sets. The Analysis Taxi is now the tool which allows any collaborator to process any data set taken since 2003 in weekly passes with turnaround times of typically three to four days.

1. The PHENIX Experiment

The PHENIX experiment [1] has been taking data since the year 2000, running between 20 and 25 weeks per year. Continuing improvements of the data acquisition system pushed the event rate to a maximum of 5-6 kHz, corresponding to about 800 MB/sec for Au+Au collisions. Run 10, taken in 2010, produced for the first time a dataset exceeding 1 PB. To expedite the reconstruction of the raw data, calibration constants which are precise enough for the pattern recognition are determined by automatic procedures within a day after the data was taken. The reconstructed output contains enough information to allow for more refined calibrations to be applied during the analysis stage, so that reprocessing of the raw data is not required to apply improved calibrations. The physics program of PHENIX covers heavy ion and polarized proton collisions. The need to combine data sets from the proton runs, as well as the large variety in species and collision energies for heavy ion runs, result in many unique data sets which have to be available for analysis over extended periods of time.

2. Data Volume, Reconstruction and Analysis

Fig. 1 shows the raw data volume collected by the RHIC experiments for each running period since 2002. PHENIX, which has steadily increased its data acquisition bandwidth, collected its first large dataset in 2004. Fig. 2 shows the number of events which are available for analysis in the reconstructed output. The improvements in terms of data rate are reflected in the increasing number of events (though one should not compare the numbers on an absolute scale given the varying the length of the runs). Fig. 3 gives the total volume of the reconstructed output that is directly used for the analysis. The heavy ion data sets, especially from Au+Au collisions, clearly dominate the data volume. The volume of our reconstructed Au+Au data sets (Run 4: $\sim 1 * 10^9$ events, Run 7: $\sim 4 * 10^9$ events and Run 10: $\sim 8 * 10^9$ events) does not scale with the number of events. It reflects the result of continued efforts to reduce reconstructed data volumes. For Run 7, we took the simple approach of only storing tracks and clusters above an energy threshold, which reduced the size by a factor of 3 compared to Run 4. For Run 10, we

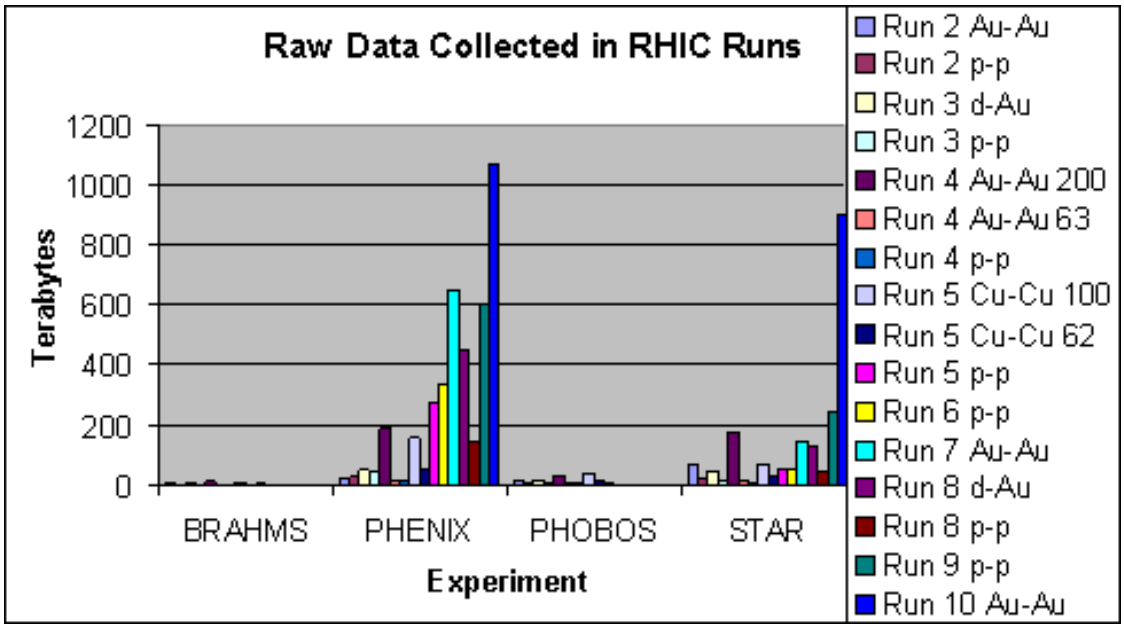

Figure 1. Raw data volume collected by the four RHIC experiments. Continuous increases in the bandwidth of the PHENIX data acquisition system lead to ever increasing data sets. The volume of yearly data sets is expected to stay in the PB range for the coming years. Incidentally, the RHIC run number corresponds to the year in which is was taken (Run2 in 2002, Run3 in 2003, etc.)

revised our output strategy again. We dropped variables which can be computed from other more basic variables and stored most of the floating point variables in IEEE 754 half precision binary floating-point format. This enabled us to remove the cutoff we had applied in Run 7 and still keep the same reduction in size. In fact we now store more information in our output than we did in Run 4. The recovery of the original data structures is accomplished as part of the calibration procedure applied at the analysis stage and is therefore transparent to the analysis software.

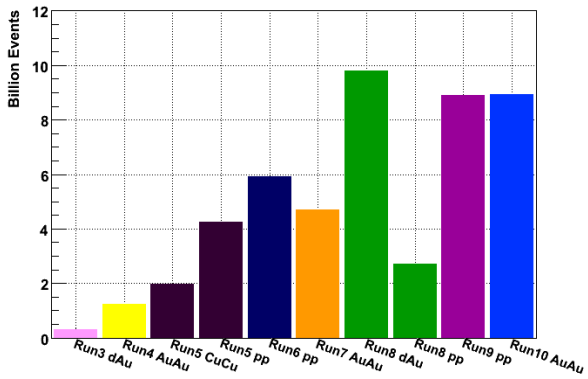

Figure 2. Number of events which are available for analysis.

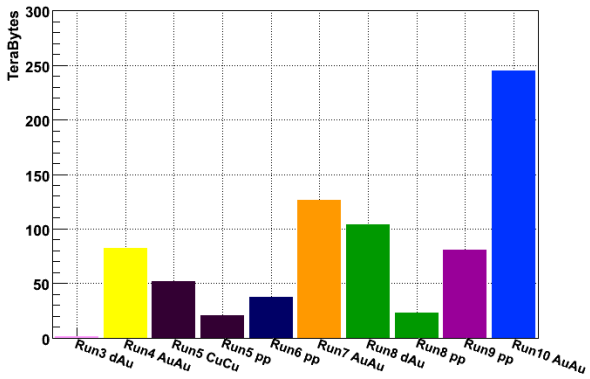

Figure 3. Total size of the reconstructed output

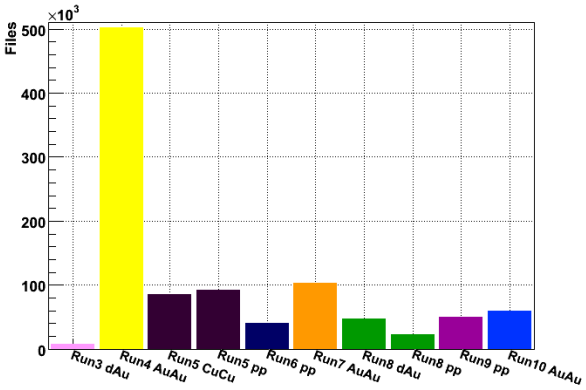

Figure 4. Number of files for each data set. Since Run 5 we aggregate the output of multiple raw data files into a single larger output file. This reduces the number of files drastically, keeping it at a manageable level.

Initially all of PHENIX's output was saved as a ROOT TTree in a single file for each raw data file. For the analysis of the Run 3 data sets, we wrote multiple output streams covering various physics topics and implemented a synchronization scheme which allowed the reading of the same event from multiple input files, satisfying the needs of analyses that required the contents of several output file streams. This approach's limitations became apparent in Run 4, as demonstrated in Fig. 4, which shows the number of output files in the data set (4 different types were produced). Even taking into account that most analysis used only one or two file types, keeping track of the analysis of 250000 files is a real challenge for the average user. To alleviate this issue, future productions involved an aggregation step where multiple files of a given type are combined until their size reaches an upper limit (starting with 2 GB in Run5, and since increased up to 10 GB). This brought the number of files down to a more manageable level. It also vastly improved the speed of staging from tape which is dominated by the number of files to stage rather than their average size.

3. From the Analysis Train...

Once PHENIX was confronted with its first large data set in 2004, it became obvious that analyzing the bulk of the data required a coordinated effort. The volume of reconstructed data was higher than that available to the collaboration in the form of centralized disk space. However, the internal disks on the compute nodes at the computing facility provided large amounts of unused disk space. This observation led to the initial Analysis Train implementation. Users interested in running over the full Run 4 data set signed up for a pass over the data with their analysis modules. The required input files were then staged to the local disks from tape storage. Once the files were staged (as much of the data set as space allowed), all analysis modules were run over the data, which were then deleted to free up space for the next portion of the data set. Every module was run as a separate process to prevent any possible interference between modules (e.g. one module modifying data structures, or corrupting the memory, thus changing results of another module). The Analysis Train approach worked [2], but the staging of files from tape was very time consuming. The turnaround time was typically one month, which was not satisfactory for most users. Also, reprocessing of failed jobs was not an option. Increasing the number of nodes to have sufficient local disk space to keep all data disk resident helped speed up the processing, but managing the content of each local disk (and with it the recovery from disk failures) was tedious and the scheme relied on all compute nodes to be always available to analyze the complete data set.

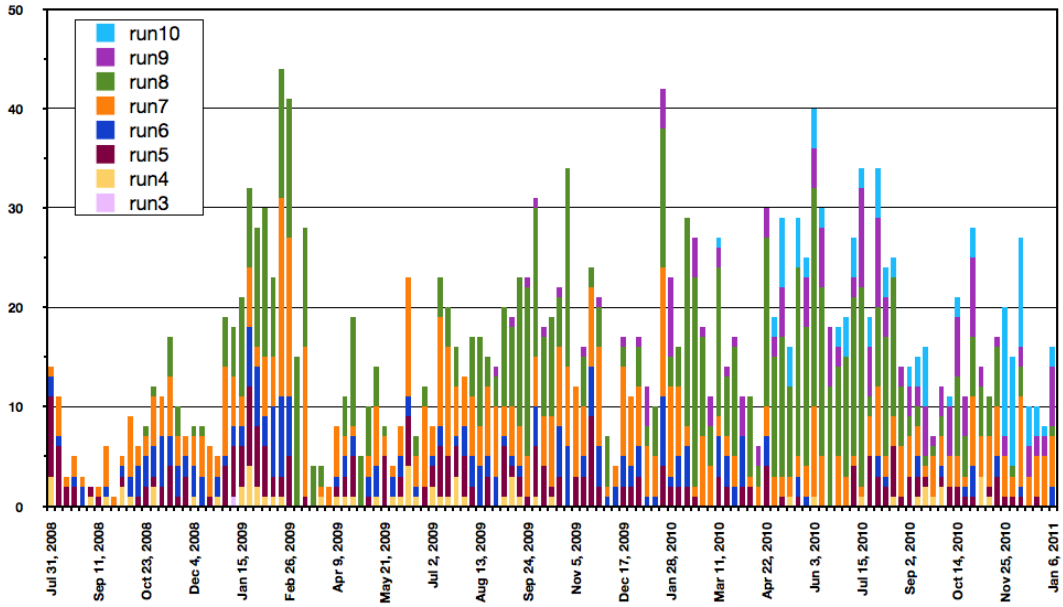

Figure 5. Number of module requests for each data set in the weekly Analysis Taxi passes. Run 10 data became available before the run was finished.

4. ...To the Analysis Taxi

The Analysis Train approach was based on collections of analysis modules and thus didn't provide the means for handling and tracking the progress of the individual analysis modules. Other drawbacks were the long time needed to complete a single pass, and the large effort it took to run. To overcome these issues, we started in 2005 to develop the Analysis Taxi, which centers on each individual analysis module. It keeps track of the module's progress in a database and reruns failed jobs until 100% of the data has been processed by the module. In addition, modules can be removed from an ongoing pass over the data to avoid wasting resources when there are problems. To ease the management of our local disk space we deployed dCache[4], and keep files which are commonly requested pinned on disk. In order to have a stable environment we decided against reading the data directly through dCache, instead adding the additional step of copying input files to the local disk of the processing node. This insulates the jobs from external problems once the data is copied. If a given job's input files cannot all be copied entirely because at least one of them is only on tape, the files are pre–staged to dCache and the job quits. The next pass over the data will then find all files available and process them. Fig. 5 shows the history of the weekly Analysis Taxi passes over the last 2 years. We typically run $\sim$ 20 new modules per week, but see sharp increases shortly before major conferences (e.g. QM2009 in April 2009). Most of the time we are capable of processing all available data sets on a weekly basis and even 6 years after the data was taken, Run4 data is still actively analyzed. Fig. 6 shows the volume of data resident in dCache (blue) and retrieved for analysis (red) on a monthly basis since 2009. The addition of about 250 TB of reconstructed data from Run 10, and their ongoing analysis, lead to a near doubling of the monthly data transfer from 1-1.5 PB to over 2.5 PB. The large transfer in February 2009 was caused by a software bug that forced the jobs to read data directly from dCache instead of the local disk. Already at that time our infrastructure was capable of sustaining this level of throughput, which gave us confidence in our ability to handle the Run 10 data via the Analysis Taxi. When the Analysis Taxi is running the overall network traffic in our computing farm is around 6 GB/sec. Fig. 7 shows the categories of

condor jobs being run on the PHENIX portion of the RACF computing facility[3]. The Analysis Taxi jobs are marked in yellow. On most weeks, the first pass over all requested data sets is finished within 3 days, leaving enough time for another pass over the data to re–process failed jobs.

5. Operation

Users can sign up for a pass via a web form. The analysis code has to be checked into our code management system, which is tagged and built for every pass. Additionally, code tests with valgrind[5] and insure[6] are required, in an attempt to detect code issues before they can create massive failures on our farm. The space for the output (typically NTuples which are then analyzed interactively) is provided by the users' physics working groups on a high performance BlueArc storage system. Another web form provides the user with a summary of processed events and still outstanding files. This form can also be used to stop the processing of a module when problems are found, avoiding the waste of resources. The module information displayed in this form is saved and can be used to document the exact input of an analysis when it is ready for publication.

6. Conclusion and Outlook

We described a system—the Analysis Taxi—which together with its predecessor, the Analysis Train, has been used for the last 5 years to analyze data taken by PHENIX. It enables us to analyze every data set we have taken since 2003 in weekly passes. It relies on $\sim 2\,\mathrm{PB}$ of low cost local disk storage managed by dCache. Currently the monthly data transfer volume is of the order of 2.5 PB, without noticeable performance issues. The use of databases minimizes the effort to run and maintain the Taxi, making it also easily portable to remote facilities. Given that the amount of local disk space keeps growing more rapidly than our reconstructed data volume, it is expected that the Analysis Taxi approach will serve as the tool for large scale data analysis in PHENIX for the foreseeable future. Users benefit from not having to deal with a huge number of input files to keep track of, and the automatic reprocessing of failed jobs.

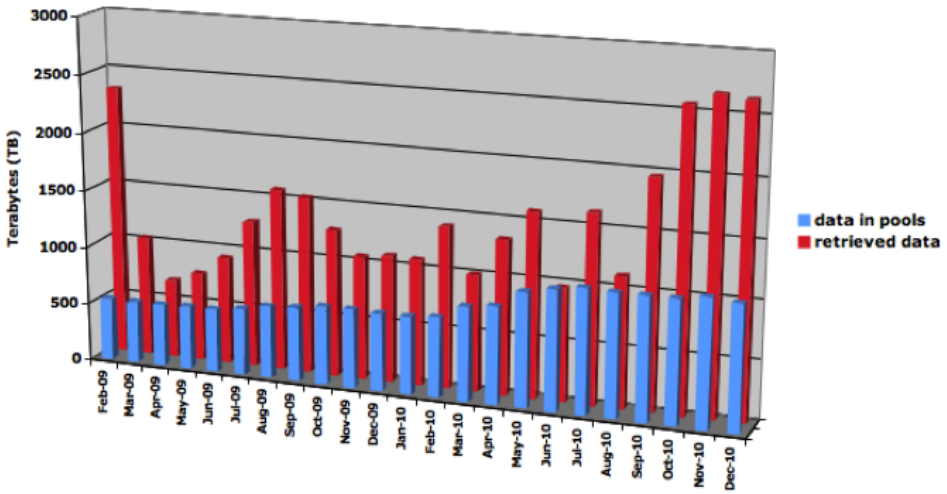

Figure 6. Monthly data transfer from dCache into the Analysis Taxi. The analysis of the Run 10 data, which started in earnest in Sept. 2010, nearly doubled the previous rate.

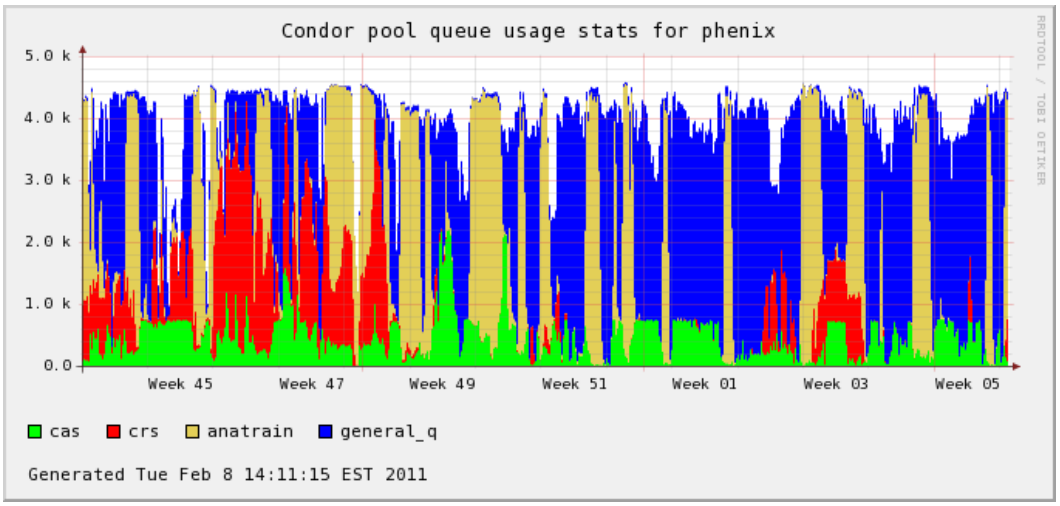

Figure 7. Categories of jobs on the PHENIX portion of the RHIC computing facility. The Analysis Taxi jobs are marked in yellow.

PHENIX's experience shows that data sets from over 5 years ago continue to be processed on a regular basis, and keeping old files readable across operating systems and ROOT versions is a major concern. The reprocessing of files which are lost due to broken/unreadable tapes is a real issue—legacy machines running the old operating system and production libraries are used to reprocess the raw data. The output of these production jobs is compared to other existing files to make sure their content is identical before the lost file is replaced. The increasing number of cores in a single compute node will eventually pose a challenge for the efficient reading of data from local disks, which are shared among cores. Preliminary tests with solid state disk showed promising results, but solid state drives need to become more cost effective before they can be deployed on a large scale.

7. Acknowledgments

The author would like to acknowledge the tremendous support by the staff of the RACF computing facility. Without their effort setting the Analysis Taxi up and running it would not be possible.

8. Bibliography

[1] K. Adcox et al. (PHENIX), NIM **A499**, 469 (2003)
[2] M. Purschke Proceedings of International Conference on Computing in High Energy and Nuclear Physics (CHEP 2006)
[3] https://www.racf.bnl.gov
[4] http://www.dcache.org
[5] http://valgrind.org
[6] http://www.parasoft.com