

4 Requirements for a New System

4.1 General

The following section states requirements that a new system should provide. These requirements are general, that is they apply to all domains.

4.1.1 Scalability

The system should be able to keep up with increasing data rates and provide required scalability. Specifically, this means that:

- PR should keep up with a peak luminosity.
- REP, SP and analysis should keep up with integrated luminosity, i.e. they must be able to re-process/re-analyze all previous year's data plus process/analyze this year's data during this year.
- A single analysis job should *technically* be able to run over an entire data set, although this will probably not be a choice made for practical reasons.
- Analysis farm should scale up to hundreds of simultaneous jobs analyzing the data.

4.1.2 Cost and Resource Usage

- The system should efficiently use hardware resources, including servers, disks, tapes, tape drives and local and world-wide network (e.g. it should not require more resources than current, Objectivity based system does).
- It should not require more people to run and maintain the system than the current system. The need for people with the right level of expertise should be considered.
- The cost of recovering from unexpected problems like a power outage, disk failure, machine reboot or job failure should be considered (length of outage, people-effort) and should not exceed that of the existing system.

- The system should provide a clear division between domains, such that changes specific to one domain should never affect other domains. Domains should not directly reference each other.
- Complex, templated schema should be supported.
- System inhibit: The system should allow to inhibit access to whole data or selected parts of the data to allow administrative tasks (maintenance).
- Authorization: Data should be protected by access control, for instance a user should not be able to modify production or other user's data (unless approved and explicitly configured to allow that). It should be possible to restrict administrative functions.
- Administration tools to manage efficiently the data should be available.
- All BaBar platforms should be supported
- The use of additional proprietary software should not increase the overall cost for any site beyond the current system.
- It should be possible to access data conveniently at all sites, of all scales (Tier A, B and C)
- Any custom software developed for a new system should be available via a standard BaBar release and any external software should be packaged such that the installation requires no more work than the existing system.

4.2 Event Store

- It should be possible to *transparently* load balance the data on data servers to improve access and recover from disk crashes.
- The system should provide a central, scalable catalog of all collections.
- Payload per event should not be bigger than in current system (after data duplication is taken into account). This might require supporting data compression on disk and borrowing parts of an event.

- It should be possible to borrow parts of an event from a previous processing to avoid duplicating them (e.g. as in reskimming, simulation done with 3 sequential executables or partial reprocessing)
- Ability to write multiple collections (e.g. streams), unless we completely rework our data model (not recommended).
- Easy way of placing event components in separate files, and distributing the files across file servers.
- The system should support pointer collections.
- The system should support collection iterators.
- Reading from multiple collections, or an alternative solution to allow digi mixing should be provided.
- User should deal with and manage event collections, not files (if these are not the same).
- It should be possible to split data production from analysis if needed for optimization.
- The overhead associated with transferring the data from where it is produced to make it available for analysis should be such that the time from data production to availability for analysis should not be more than one week for new data. (Reprocessed data may be blocked into larger chunks, of course, but should not be technically limited to more than one week.)
- File size to HPSS: average target size should be about 250 MB, and certainly should not be smaller than 100 MB. There is no upper limit, though it probably should not go much above 10 GB.
- System should provide an easy access to files on tapes, including a file staging mechanism, purging and migration.
- It should be possible to keep some parts of an event on disk and some on tape.
- daemon based access should be provided

- The tag structure should be flexible and easily extensible, it should not put any restrictions on types of number of attributes.

4.3 Conditions

[I guess Igor's documents for the new design can be referenced here and perhaps a short summary included.]

4.4 Configurations

4.5 Ambient

4.6 PR Spatial/Temporal

4.7 Data distribution requirements

- Data should be distributed in a way that does not require the use of particular vendor.
- Data distribution must be architecturally neutral.
- Data must be able to distributed in an unpinned way (that is, references within the data must be self-contained and be independently relocatable).
- Distribution (sending and receiving) must be able to be performed in parallel with analysis and production.
- Meta data about the data must be stable. That is, it should not be necessary to resend a lot of meta-data each time a file is transferred.
- The distribution model must be compatible with emerging grid technologies.
- It should be possible to exchange data between sites without clashes between identifiers used (database ids, file names, or whatever used).