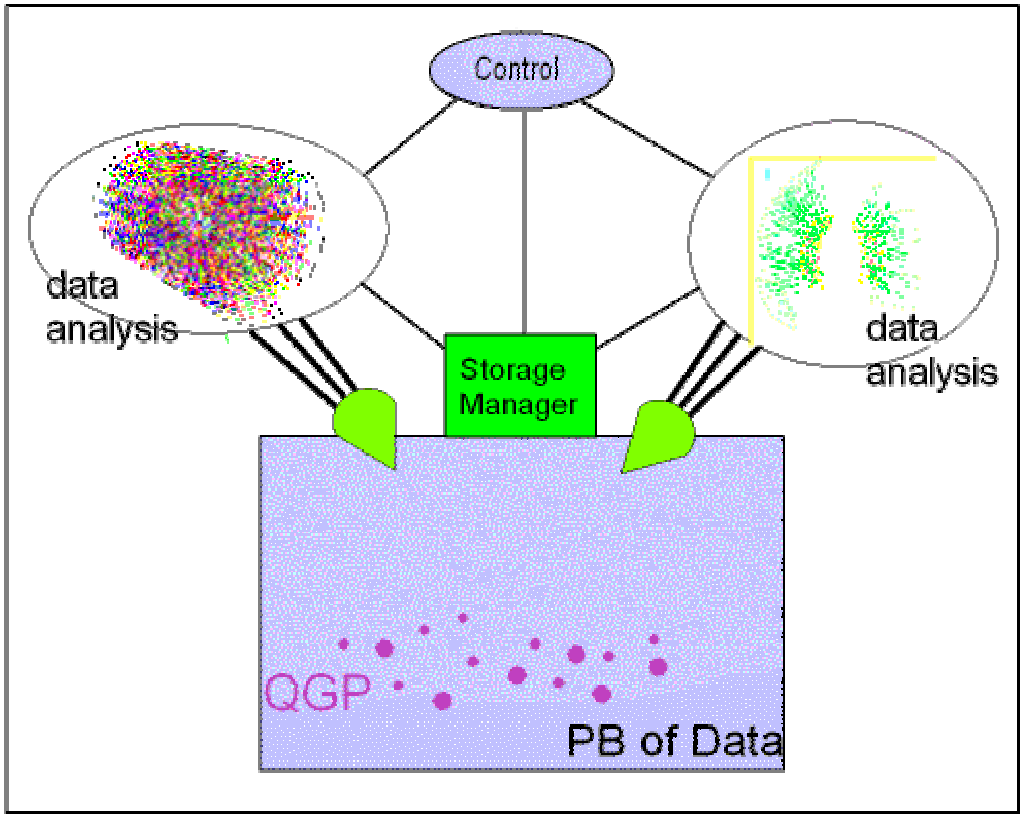
Root and the HENP Grand Challenge for Efficient Data Access



http://www-rnc.lbl.gov/GC/

David Zimmerman

for the Grand Challenge Collaboration

US Root User's Workshop March 25, 1999

Outline of Presentation

- Who are the Grand Challenge?
  - Collaborators in the HENP Grand Challenge (GC)
- Why would one want the GC architecture?
  - Motivation
- What is the Grand Challenge Architecture (GCA)?
  - Description of Software Components
  - Use-Case Scenario
- Assumptions are made about experiments which use the GCA
- Relationship to Root
- Status and plans.

GCA Collaboration

- Dave Malon, ATLAS/ANL
- Henrik Nordberg, NERSC/LBNL
- Luis Bernardo, NERSC/LBNL
- Alex Sim, NERSC/LBNL
- Jeff Porter, STAR/BNL
- Dave Zimmerman, STAR/LBNL
- Jie Yang, STAR/LBNL-UCLA-Beijing
- Stephen Johnson, PHENIX/Stony Brook
- Doug Olson - STAR/LBNL
- Arie Shoshani - NERSC/LBNL (Data Mgmt Group)
- Bruce Gibbard RCF/BNL
- Torre Wenaus STAR/BNL
- Ed May - ATLAS/ANL

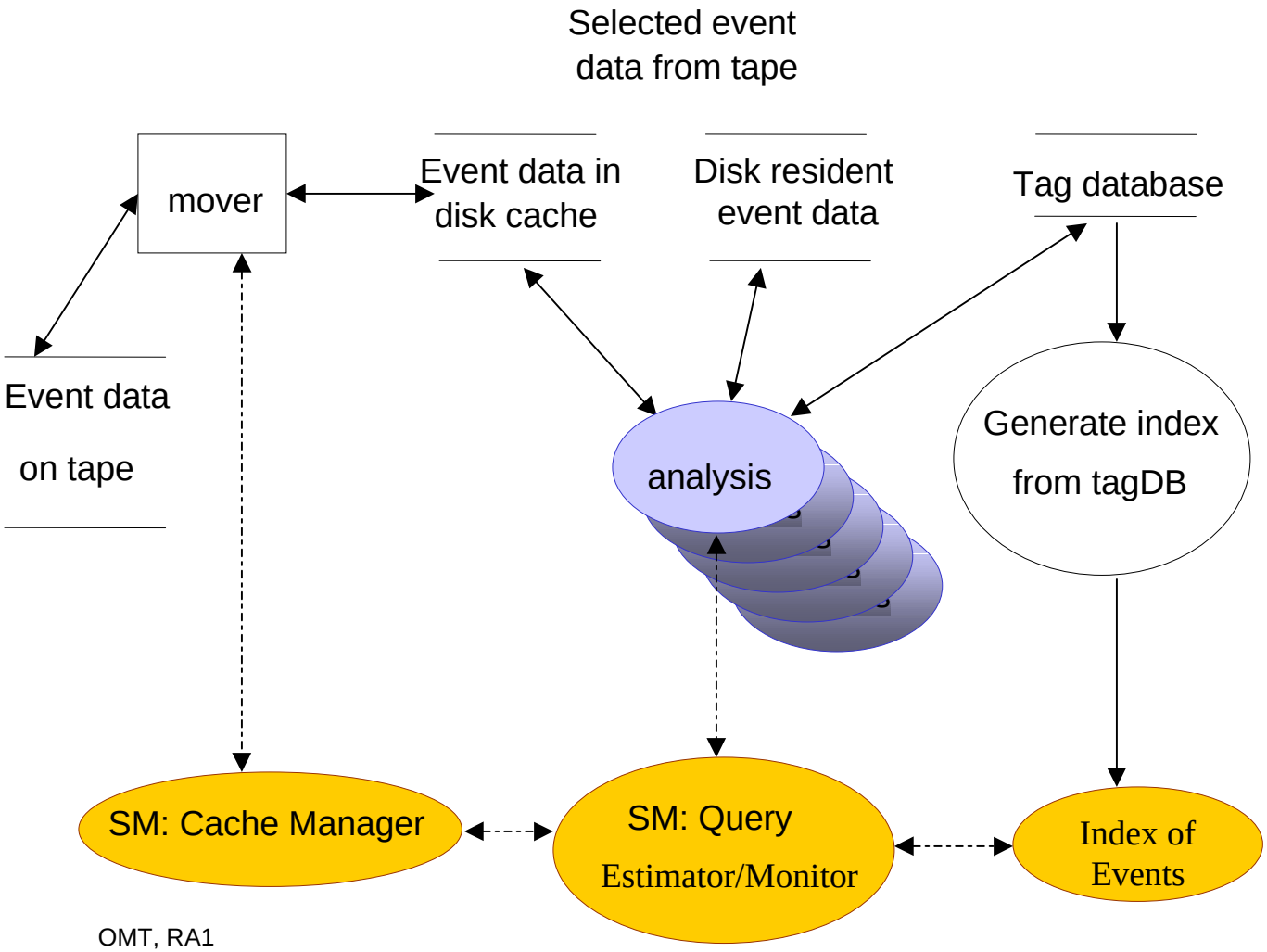
Motivation For Grand Challenge Project

- Event Stores $10^{7\text{-}8\text{-}9}$ Events per year
- 100's + of Terabytes data/year
- Bottleneck of shared access to tape store
- Complex data models
  - Individual events distributed through multiple files
- Distributed analysis and processing

GC Features

- The Grand Challenge project is concerned primarily with the tape-disk interaction. The basic mechanism is:
  - convert event list (object list) to set of files containing those events
  - pre-stage those files to disk cache
  - return sub-lists corresponding to cached files to multiple processes for parallel analysis.
- Optimize order of file fetching from tape across all active queries
- Provide query estimation to determine number of events, files, and amount of time for an analysis job before execution. Provide the opportunity to abort query prior to execution.
- Provide monitoring information so that disk & tape level clustering effectiveness can be evaluated.

Grand Challenge User Scenario



OMT, RA1
29 Mar 98

1 User's Analysis program presents query to Query Estimator (receives token).

2 Analysis processes coordinate with query monitor via query token.

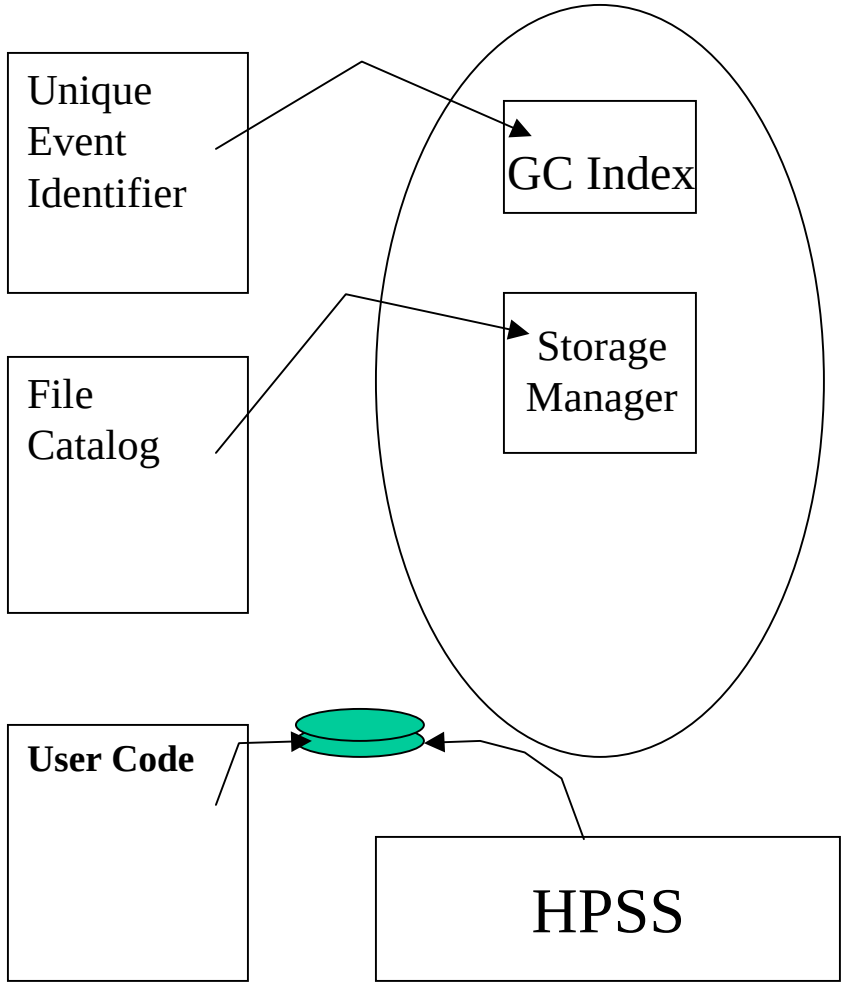
3 When user opts to execute the query a file list is passed to the query monitor.

4 Cache Manager, checks which of the requested files are already on disk and begins to pre-fetch files out of HPSS.

5 Event lists from files which are available are passed to analysis codes via the Order Optimized Iterator.

Requirements for an experiment using the Grand Challenge

- File Catalog
  - concept of a file identifier
- Tag database
  - concept of a unique event identifier
  - Queryable event attributes
  - Associated files in file bundle
- Ability to “build” a transient event based on Event ID and associated files

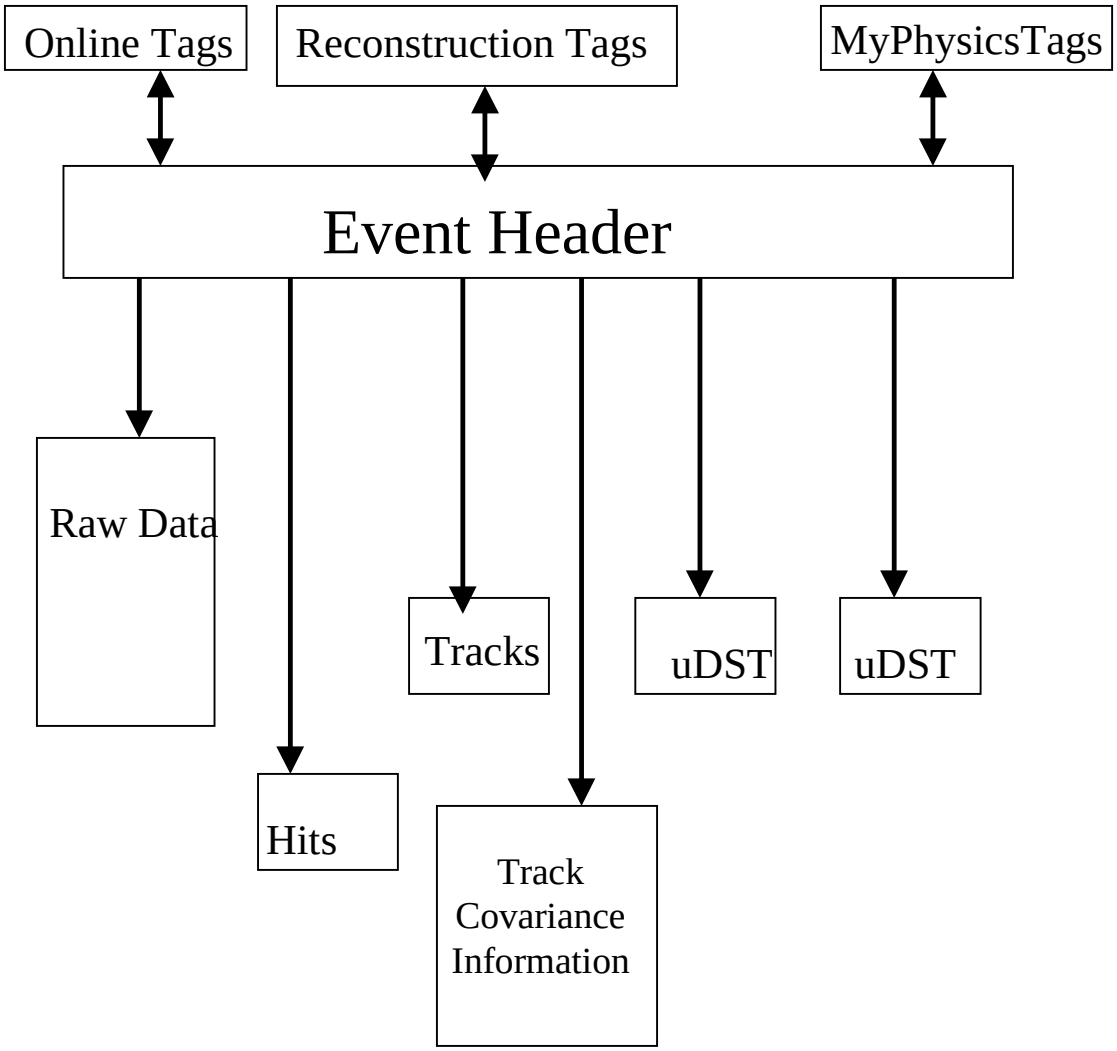


Assumptions about Event Model

- Each event has: Event identifier
  - OID of event header object (Objectivity implementation)
  - Root FileID-Sequence Number (Root implementation)
  - Multiple named components (raw, hits, tracks, ...), a given component for one event exists in one file
    - each component has top level object used for finding the whole component
  - Event Store index with queryable information and event OIDs.
    - Event header object
    - Tag object in the tagDB with attributes

Expected Event Model for GC events

Each Event Consists of an event header with references to event components which exist in various files



A Possible ROOT Event Identifier

- ROOT event-object identifier is
  file + sequence number in file
- TEventList is a list of sequence numbers which identifies the location of an object in a root tree.
- A fileID + TEventList uniquely identifies a set of events (or components) in the event store.
- What to use analogous to global event header object?

File Catalog

- Mapping of file name and location to FileID number.
- Interfaced to the GC storage manager
  - Presently we use HPSS with the “Objectivity database catalog” to locate “databases” (both Objectivity files and root files): A temporary fix
  - Soon to use Objectivity as database of file information.

Status and Plans

- Status
  - GCA in use for STAR MDC1 and MDC2
  - Integrated with objectivity based event store.
  - Exercise multi-file event component model in RHIC MDC2.
- Upcoming Plans
  - Further scalability testing in June
  - Better integration with Root
    - Interface with pure root event model
  - Objectivity based file catalog

Issues for Root (and no Objectivity)

- Equivalent of Event header object
  - unique event identifier
  - covers the event life cycle
  - provides reference to all event components throughout the event life cycle
- File Catalog