GCA Scalability June ‘99 Test Federation

D. Zimmerman

Argonne Meeting

April 28, 1999

Requirements for the Scalability test Event Store

- >=10 million events.
- Multiple components for each event, distributed across files.
- More complicated distribution of event components in files. "Multiple Bundle Overlap"
  - E.g. The number of events stored in each component file varies from component to component.
- Several hundred Gigabyte of data in HPSS.
- Event model which allows user to navigate to at least one component from the event header/
- Tag information for each event (EvID, Bundle, Queryable tag info). ***Need 25 to 50 components***

Proposed Event Store

- LBL-HPSS or BNL-HPSS
- Root based with some “RAW-format” (unreadable) components
- No permanently disk resident components
- Includes event components which are non-navigable (large, unreadable files as part of each events fundamental bundle). Similar to RAW data.
- Tags consist principally of random numbers, tagrand and FileIDs.
    - How many tags are required for tests? Do we need actual calculated tags for the events?
        - If tagdb consists of:

Struct ThisEventTag {EvOID, Comp1, Comp2, …CompN, TagRand};

Simulated Root Event

Event and Track classes

The Event class is a naive/simple
example of an event structure.

public:
Int_t fNtrack;
Int_t fNseg;
Int_t fNvertex;
UInt_t fFlag;
Float_t fTemperature;
EventHeader fEvtHdr;
TClonesArray *fTracks;

The Event data member fTracks is a pointer to a TClonesArray.
// It is an array of a variable number of tracks per event.
// Each element of the array is an object of class Track with the members:
// private:
// Float_t fPx; //X component of the momentum
// Float_t fPy; //Y component of the momentum
// Float_t fPz; //Z component of the momentum/
/ Float_t fRandom; //A random track quantity
// Float_t fMass2; //The mass square of this particle
// Float_t fBx; //X intercept at the vertex
// Float_t fBy; //Y intercept at the vertex
// Float_t fMeanCharge; //Mean charge deposition of all hits of this track
// Float_t fXfirst; //X coordinate of the first point
// Float_t fXlast; //X coordinate of the last point
// Float_t fYfirst; //Y coordinate of the first point
// Float_t fYlast; //Y coordinate of the last point
// Float_t fZfirst; //Z coordinate of the first point
// Float_t fZlast; //Z coordinate of the last point
// Float_t fCharge; //Charge of this track
// Int_t fNpoint; //Number of points for this track
// Short_t fValid; //Validity criterion

Proposed Event Store Pt.2

- 1000 Runs (files) of 10000 events each
  - 3 Root (branches) files per event
  - Individual Root event size of 50Kb
  - 3 Root files per run whose total size = 1.5 GB
  - Two 250 Mbyte “RAW” file per run
- 10^7 events in the store
- 1000 files (each event consists of 5 files)
- Total volume ~2TB

Event characteristics

- Only Root event components (branches) are navigable
- Complex File Bundles
  - More realistic to predefine the relative size of the components for an event and “fill” files (fixed to 1 gig) with events during event creation
    - eg. 200 events in RAW -> 2000 events in Trks.
  - Another possible multiple bundle overlap:
    - The association of an event with a RAW file is made stochastically
    - Where the RAW files associated with a given Event are selected from the set of RAW files via:
      - 30% chance of ThisRAWFileNumber-1
      - 50% chance of ThisRAWFileNumber
      - 20% chance of ThisRAWFileNumber+1