MeritTuple status, proposals

Elimination of now-redundant variables
Double to float (or int)
Arrays in the tuple
GEM status word bits set in MonteCarlo
LiveTime
Redundant variables

- Event-level variables set in merit are now also set in AnalysisNtuple::EvtVarsTool and McVarsTool. I propose, on an agreed-to time schedule, to eliminate these obsolete variables before DC2.

<table>
<thead>
<tr>
<th>Current (DC1)</th>
<th>Replacement (DC2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event_ID</td>
<td>EvtEventId</td>
</tr>
<tr>
<td>Run</td>
<td>EvtRun</td>
</tr>
<tr>
<td>elapsed_time</td>
<td>EvtElapsedTime</td>
</tr>
<tr>
<td>MC_src_Id</td>
<td>McSourceId</td>
</tr>
</tbody>
</table>
Why is the MeritTuple all double now?
- Basically, because all the AnalysisNtuple member variable set by the various ValsTools classes are double, and the visitor can only pass pointers to these.

What has changed?
- Leon has changed the class to handle floats and ints, and the visitor now has three callbacks.

Advantages
- Factor of two (even compressed) in storage, download times

Any downside?
- The conversion program to Insightful Miner may need to be changed to use the ROOT Value() function.

One exception: EvtElapsedTime must be double since it is copied to FT1 from the tuple.
Arrays (constant length)

- Supported by ROOT and RootTupleSvc:
  - Simply add “[n]” to the variable name, where n is a constant integer, and that is it!
  - Pointer is then interpreted as being to an array of the pointer type.

- Advantages:
  - Economy of variable names
  - Possibility of implied loop over the index

- Example: CalElayer[8] (see next slide)

- Proposal
  - Add the array definition(s) as aliases initially, consider need for specific entries (like CalELayer5).
Array test example
<table>
<thead>
<tr>
<th>Bit</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ROI</td>
<td>Meaning depends on whether ACD is being used as veto or trigger</td>
</tr>
<tr>
<td>1</td>
<td>TKR</td>
<td>OR of 3-in-a-row for each tower</td>
</tr>
<tr>
<td>2</td>
<td>CAL (LE)</td>
<td>OR of CAL low energy for each tower</td>
</tr>
<tr>
<td>3</td>
<td>CAL (HE)</td>
<td>OR of CAL high energy for each tower</td>
</tr>
<tr>
<td>4</td>
<td>CNO</td>
<td>OR of 12 ACD CNO inputs</td>
</tr>
<tr>
<td>5</td>
<td>Periodic</td>
<td>set for periodic trigger</td>
</tr>
<tr>
<td>6</td>
<td>Solicited</td>
<td>set for solicited trigger.</td>
</tr>
</tbody>
</table>

For MC events, we are now setting the bits in the tuple word GemConditionSummary corresponding to TKR, CAL(LE), CAL(HE) and CNO.
Live Time

- New temporary variable: \textit{LiveTime}
  - Will be named \textit{EvtLiveTime}, if/when it is eventually set in AnalysisNtuple::EvtValsTool.
  - Definition: cumulative live time up to current event
- MC implementation, in TriggerAlg
  - Add elapsed time from previous trigger, less the value of \textit{TriggerAlg.deadtime}.
  - If an event occurs sooner than this, do not set new trigger bit. (is there ever a reason to keep such an event?)
- LDF implementation
  - Not set yet.
Plot example

- Set TriggerAlg.deadtime to 0.1 s (like EGRET), require pass dead time trigger condition:

  Shows the accumulated deadtime increasing with event id. Total is 83 s corresponding to 829 triggers.