art - Feature #22403

Support an option for a wall clock time limit for the EmptyEvent source

04/18/2019 12:06 PM - Rob Kutschke

Status:	Closed	**Start date:**	04/18/2019
Priority:	Normal	**Due date:**	
Assignee:	Kyle Knoepfel	**% Done:**	100%
Category:	Infrastructure	**Estimated time:**	0.00 hour
Target version:	3.03.00	**Spent time:**	4.00 hours
Scope:	Internal	**SSI Package:**	art
Experiment:	Mu2e		

Description

The HPC resources used by Mu2e use whole node scheduling - that is we get the whole node for the requested time or until we are finished with it, which ever is earlier. On a typical KNL node we run something like 32 processes with 8 threads each or some other variation that adds up to 256 threads.

Our big CPU driver is for stage 1 MC jobs that use the EmptyEvent source. Currently we submit these jobs requesting a fixed number of events in each job. There is a big dispersion of execution times for jobs with a fixed number of events. Suppose we submit jobs that we expect will have a mean duration of 4 hours and a tail to 8 hours. On a typical node the first process might end after 3 hours and the last after 6 hours - so we have just wasted 1/64 of the allocation (1 of 32 processes for half of the overall time).

Over an ensemble of jobs, I bet this averages to about 25% of the total available cycles. Chris Jones has told me that CMS is getting pushback from the HPC centers about this.

For any job that uses EmptySource, we can choose a different strategy. We can tell the job to run for a fixed time. For example we might submit jobs with a time limit of 6 hours and tell art to stop processing events after 5:30 or 5:45. We are not charged for the time that is left over after the last process exits so it's not critical to do a detailed optimization of this backoff.

We request that art provide an option on EmptyEvent to tell the job to run until it has used a fixed amount of wall clock time. If it is too expensive to check the elapsed wall clock time after every event, then please provide an option to check the elapsed time every N events. We prefer that it be a configuration error to provide both a maximum wall clock time and a maximum number of events. For jobs that use EmptyEvent this would mean only a modest change in our workflow management and bookkeeping.

At this time we are not interested in this feature for RootInput since that would require a very intrusive change in our workflow management and we don't need that feature at this time. It's possible that we might request this feature in RootInput at a later date - but I hope not.

Rob

History

#1 - 04/18/2019 12:15 PM - Kyle Knoepfel

- Description updated

#2 - 04/22/2019 10:27 AM - Kyle Knoepfel

- Status changed from New to Accepted

This is a reasonable request. Please let us know a timeframe when you would like this feature implemented.

#3 - 04/22/2019 10:32 AM - Rob Kutschke

We would like it to be available by June 1 , 2019.

#4 - 05/14/2019 10:43 AM - Kyle Knoepfel

- Target version set to 3.03.00

#5 - 05/17/2019 11:07 AM - Kyle Knoepfel

- % Done changed from 0 to 100

- Assignee set to Kyle Knoepfel

- Status changed from Accepted to Resolved

- Category set to Infrastructure

- SSI Package art added

This feature has been implemented with commit art:13eea62b. An additional parameter called maxTime has been added to the EmptyEvent configuration. Per stakeholder discussion, the maxTime parameter cannot be used with either maxEvents or maxSubRuns. Attempting to do so will result in a job-ending exception throw.

The maxTime value corresponds to the maximum number of seconds the EmptyEvent source is allowed to construct new events, subruns, or runs. The clock begins at EmptyEvent construction time, and not the beginning of event-processing--although this difference may not be significant in most cases, it is one to be aware of, especially since the time report at the end of each *art* job corresponds to the execution of the event loop.

Note also that although new events will not be created after the maxTime value has been exceeded, the processing of the last event will continue uninhibited. This means that jobs will likely take longer than the maxTime value specified.

#6 - 06/26/2019 09:22 AM - Kyle Knoepfel

- Status changed from Resolved to Closed