UCRL-SM-228839

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Options for Retrieving Remaining Time Under Moab

Donald A. Lipari, Christopher J. Morrone

March 9, 2007
Updated March 19, 2012

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory in part under Contract W-7405-Eng-48 and in part under Contract DE-AC52-07NA27344.

Introduction

Users will typically write code to run for a set period, write a checkpoint file or save state, and then exit gracefully before the job's wall clock limit is reached. There are two basic ways for a running job to discover when its wall clock limit is about to be reached. It can either request a signal from the scheduler at a certain time prior to its wall clock expiration, or periodically call an API that returns its remaining time.

Requesting a Signal from Moab

Those who prefer that their jobs receive a signal as it approaches its time limit can specify that request as part of the `msub` invocation:

```
msub -l signal=<sig_num>[@<secs_remaining>]
```

The user must write a signal handler to catch the signal Moab will send when the job has `secs_remaining` seconds left to go.

Polling for Remaining time from Moab or SLURM

There are three separate API's for retrieving a job's remaining time. Each has their own advantages and disadvantages. All three follow this basic pattern:

```
while (work) {
  if (get_remaining_time() < gracetime) {
    save_state();
    exit(0);
  }
  do_work();
}
```

Here is a summary of the three APIs available in order of recommendation.

The Yogrt Library

For users who want high accuracy and tri-lab compatibility, a custom API has been provided, the yogrt library (see "man libyogrt" for details). `yogrt_remaining()` will return the number of seconds remaining in the job allocation.

`yogrt_remaining()` is designed to be fast, with internal caching of the remaining time reported by the resource manager, so calling it relatively often should not result in negative performance impacts.

SLURM's API

For jobs running on machines controlled by Moab / SLURM, SLURM's native API, `slurm_get_rem_time()`, provides a simple and very accurate method for determining when a job's time is about to expire.

Moab's API

For users running jobs on any of the Tri-Lab Moab machines, Moab's native API, `MCCJobGetRemainingTime()`, provides the remaining time, no matter what the underlying resource manager is. It is documented in Appendix H of the Moab Workload Manager Administrator's Guide. While accurate to within a minute, this library call is not as accurate or as responsive as `yogrt_remaining()` or `slurm_get_rem_time()`.

Summary - Scenarios

The following tables summarize the recommendations for users running jobs under Moab at LLNL and at the other Tri-Lab facilities.

Requesting a Signal and Writing a Signal Handler:

Compiled for machines running under…	**Use**
Moab at LLNL	msub -l <sig_num>[@<secs_remaining>] or psub -S <sig_num>[@<secs_remaining>]
Moab at any Tri-Lab	msub -l <sig_num>[@<secs_remaining>]

Polling for Remaining Time

Compiled for machines running…	**Use**
Moab at LLNL or on any TLCC2 cluster	`yogrt_remaining()`
Moab/SLURM	`slurm_get_rem_time()`
Moab at any Tri-Lab	`MCCJobGetRemainingTime()`