

Resource manager needs from checkpoint/restart perspective

Adam Moody

moody20@llnl.gov

Jan 12, 2017

Queries for basic info

- Static values
 - Job id
 - Job name (string assigned by user)
 - Cluster name (not hostname, but machine name)
- Dynamic
 - Time remaining in job allocation (e.g., libyogrt)
 - <https://github.com/LLNL/libyogrt>
- Job-specific resources
 - Directory paths to temporary storage allocated just for job
 - Allowed capacity (how much of SSD can the job use)

Heartbeat

- Provide mechanism for job to tell resource manager next time it should check in
- Assume job is hanging (and thus should be killed) if it fails to report in given time
- E.g., set heartbeat at 3600 seconds
 - PMIX_Put('heartbeat', '3600');
 - PMIX_Put('heartbeat'_exec, 'path/to/script');
- With iowatchdog, it's useful to let user specify a script to run if timeout is triggered
 - e.g., LLNL captures stack trace, emails user, kills job step
 - <https://github.com/grondo/io-watchdog>

Job launch / control from job script

- Uniform method to start an MPI job
 - `srun -n 2`
- Uniform method to kill a running MPI job
 - `scancel <jobstep.id>`
- Uniform method to lookup jobid, given job name?

Node health and replacement

- Query resource manager for known bad nodes
- Run application-specific node-health check on supposed good nodes
 - E.g. in SCR, we test nodes with ping and run check scripts on compute nodes with pdsh
- Report failed nodes to resource manager
 - Often MPI detects a failed node before the RM (slurm node health check runs every 5 minutes, but MPI detects failed node immediately)
 - Application (or user) may detect a failed node which is not detected by RM tests
 - e.g., in SCR we can't report, but we avoid bad nodes with srun -x
- Request spare node for allocation
- Specify node set when launching a job (srun -w)
 - Run next job with maximal overlap of node set from previous job, since this node set holds most recent checkpoint

Communication between job and job script

- Enable job script to Get key/value pairs put by MPI job

```
while [ 1 ] ; do
    srun -n2 myprog
    if [ pmix_get 'done' ] ; then
        break;
    fi
done
```

- Need mechanism to flush/sync keys to know they're visible to job script
- Would like to flush keys written by job at job failure
- Similarly, job script may want to Put key/values to MPI job

External control

- Enable users to submit keys for job while queued or running via login node which job and/or job script can read at runtime
- Users can steer running simulation