Layout for checkpoint restart on specialized blades

Slurm User Group 2018

Bill Brophy
Martin Perry
Doug Parisek
Steve Mehlberg

26-09-2018
Table of contents

- Motivations
- PCIe switches and FTI
- Allocation and restart
- Conclusion
Motivations
Motivations
Why checkpoint restart

- Increasing HPC systems size => increasing allocations size
- A single (hardware) failure may affect huge applications and loss of hours of run
- Checkpoint and restart well-known
  - need shared filesystem
  - overhead on implementation
  - overhead on run
  - restart mechanism needed
- Target: ease of use for final user
Motivations
Atos: hardware and software provider

- Specialized blades for Bull Sequana
- Open-Source and well known checkpoint-restart tool (FTI)
- Dedicated software for checkpoint/restart
  - Adding a Layout to Slurm
  - Scripts for blades management
PCIe switches and FTI
PCIe switches and FTI
Sequana Blades

- 3 nodes per blade
  - Management
  - Redundant
  - no role
- 3 disks per blade
  - connected to 1 node
- PCIe switch
  - Peripheral Component Interconnect Express
  - controlled with Bull PCIe Switch Management
  - info, activate, migrate/grab
PCIE switches and FTI

FTI

- Fault Tolerance Interface
- Multi-level checkpoint/restart library
  - 4 levels
    - local (SSD, fastest, low reliability)
    - neighbor (replication, fast copy, tolerates single node crashes)
    - Reed-Solomon shared data (Encoded, quite slow, very reliable)
    - parallel filesystem (Slowest, most reliable)
  - API
    - Init/finalise (need MPI)
    - Protect (defined pointer to protect)
    - Checkpoint level, recover
    - ...
PCIe switches and FTI

FTI

- Node reordering
  - usefull when restart
  - MPI_Comm_World => FTI_Comm_World
- Dedicated post-processing process
  - limit overhead (on non local checkpoints)
- Protected variables updates
  - support reallocation (moved memory, increased size)
- Uniq Id to restart from checkpoint

- Focusing on level 1
3 Allocation and restart
Allocation and restart
Hardware needs software

- Blade aware allocation
- Recover data from failing nodes
Allocation and restart
Architecture inside Slurm

- Needs
  - to be aware (and use) relationship between PCIe switches and nodes
  - to be aware of roles of nodes
  - to be aware of previous allocation for restarting step
  - propagate failure information in epilog scripts

- select plugin based on cons_res
  - contains PCIe aware node selection
  - performs reselection when job Restarts
  - only used if configured in slurm.conf & designated by the job
  - also contains additional Atos/Bull specific enhancements

- PCIe description by a dedicated layout
Allocation and restart
Slurm Layouts Framework

From CEA slides of Slurm User Group 2015 (Matthieu Hautreux)

- Started in 2012 and introduced in 14.11

- Goals
  - Add a generic/extensible way to describe facets of supercomputers
  - Propose facets details to the resource manager for
    - Advanced management
    - Advanced scheduling
  - Ease facets information update to take into account system dynamics
Allocation and restart
New layout

- **Type=Center|Switch|Node**
  - Center is the type for the Cluster entity
  - Switch is the type for a PCIe switch entity
  - Node is the type for a compute node entity connected to a PCIe switch.

- **Enclosed=<nodelist>**
  - `<nodelist>` (for switches) is the list of compute nodes connected to this PCIe switch entity.

- **Role=Manager|Backup|Other**
  - Manager, Backup and Other are the three possible roles for a node on a PCIe switch
  - Each switch must be configured with one node for each role.
Allocation and restart
New layout example

- etc/layouts.d/switch.conf
  Priority=10
  Root=Cluster Type=Center Enclosed=PCle[0-1]
  Entity=PCle0 Type=Switch Enclosed=node-[0-2]
  Entity=PCle1 Type=Switch Enclosed=node-[3-5]
  #Node level Layout configuration
  Entity=node-[0,3] Type=Node Role=Other
  Entity=node-[1,4] Type=Node Role=Manager
  Entity=node-[2,5] Type=Node Role=Backup
Allocation and restart
Submission and environment variable

- `sbatch [--checkpoint_restart] --exclusive [<other options>]` script
  - only `sbatch`
  - needs (forces) exclusive
  - compatible with everything (almost... `-w/--nodelist`)

- Environment variables
  - `SLURM_PREV_NODES` – names of nodes previously allocated to job (new)
  - `SLURM_RESTART_COUNT` – indicates restarting
    - existing but exported in epilog
  - `SLURM_JOB_DOWN_NODES`
Allocation and restart

Allocation rules

► With checkpoint restart option
  – Allocate at least the Manager AND the Backup nodes of the same blades
  – Allocate a maximum of full blades
  – Fail if not possible (wait in queue)

► Without checkpoint restart
  – prefer the not complete blades and other nodes

► Restart
  – reallocate other blade(s) to replace if possible
  – if not possible, requeue (as usual checkpoint-restart)
**Allocation and restart**

**Epilog**

- Bash script in job epilog, run on all nodes
  - Am I a “checkpointable” job
  - Does the job fail?
    - check `SLURM_JOB_DOWN_NODES`
  - is blades concerned?
    - check `SLURM_JOB_DOWN_NODES ∩ SLURM_JOB_NODELIST`
    - I’m the Manager
      - I take control of the disk from the down node (API)
    - I’m the Backup & the down node is the Manager
      - I take control of the PCIe and I do as previous (API)
  - Move the last checkpoints to persistent file system
Conclusion
Conclusion
Validation

- Slurm 17.11
- Internal functional validation
- External validation with cooperation with CEA
- Scalability on simulated (multiple-slurmd)
  - 5040 nodes
  - light-ESP workload
  - comparison with/without configuration and option
Conclusion
without

- No layout
- No specific option
- No failure on nodes (no restart)
Conclusion with

- Layout configured for all nodes
- All jobs started with the new option
- No failure on nodes (no restart)
Conclusion
To the community

- Behavior and new environment variables

- Layout very Atos hardware dependent
  - need to be more generic (blades of 3 nodes)
  - need to generalize roles (manager, backup, other)

- Adapt to cons tres ?
Thanks

For more information please contact:
thomas.cadeau@atos.net