

Checkpoint/restart in Slurm: current status and new developments

M. Rodríguez-Pascual,
J.A. Moríñigo, R. Mayo-García

Outline

- Checkpoint/restart
- Integration in Slurm
- Our roadmap

Checkpoint/Restart System vs. Application level

Characteristics	System Level	Application level
Triggered by:	User/system	Application
Basic idea	Full memory dump	Save relevant information
When to checkpoint?	Any time	Pre-fixed places
Requires modification of application	No (some technologies require re-compilation)	Yes
Resulting file size	Big	Small
Overhead in exec. time	~1-2%	negligible

- This work is about system-level checkpoint: application level one is transparent to Slurm

BLCR Vs. CRIU Vs. DMTCP

- Disclaimer: this is our personal experience

	CRIU	DMTCP	BLCR
Integration with Slurm	NO	NO	YES
Requires application modification	NO	NO	Recompile app
MPI applications	NO	YES	YES
Can checkpoint running application without preloading	YES	NO	YES* library must be loaded
Overhead besides checkpoint	NONE	Init: sec. CPU: 1-2%	CPU:1-2%
Can checkpoint containers (Docker & LXD)?	YES* we have only tested Docker, not LXD	NO	NO
Infiniband support	N/A	YES	NO* we haven't tried, comes from doc.

Random thoughts

- DMTCP & CRIU do not oblige to recompile: useful for legacy & proprietary applications
- CRIU & BLCR do not oblige to start anything before the application
- DMTCP & BLCR can checkpoint MPI
- CRIU can checkpoint Docker & LXD containers
- (Small) overhead with CRIU & BLCR
- Future plans?

- ... different solutions depending on your needs

What have we done?

- Implement Slurm drivers for DMTCP & CRIU
- Add “--with-criu=<path>” and “--with-dmtcp=<path>” flags on compilation
- Add “--with-criu” and “--with-dmtcp” flags to sbatch

CRIU Plugin

- Direct adaptation of existing BLCR one
- Idea:
 - Do nothing when starting the application
 - On checkpoint, execute a script
 - Organize data: input, folder structure and so
 - Get process PID
 - Call CRIU checkpoint op.
 - On restart, execute another script
 - Call CRIU restart op.

DMTCP Plugin

How does DMTCP work?

- DMTCP employes a coordinator
 - All jobs register there
 - It coordinates checkpoint on parallel applications
 - Can be more than one. We are employing one per node (max.)
- Coordinator must start before application
- Application is started trough a wrapper

DMTCP Plugin

Checkpoint process

- The whole script submitted to sbatch is checkpointed.
- When running a job, see if there is a coordinator in that node.
 - If so, connect to it. If not, start one
 - When running a multi-task job, only one task performs this. The rest connect to the coordinator

DMTCP Plugin

implementation (1): Checkpoint/restart

- Performed with a script, just like BLCR one
- Only difficulty is to know where is the coordinator running.
- All together, quite straightforward.

DMTCP Plugin implementation:

Implementation (2): job start

- Performed with a Spank plugin.
- We capture the command to be executed on the remote node (user script). Then we replace it by a call to the DMTCP wrapper, with the user script as input argument
- On start, this wrapper
 - Includes a mutex, so only a task (in a parallel job) executes it
 - Organizes some files, folders and so
 - If there is a coordinator running, connects to it. If not, start some
 - Starts the real job to be executed (which is the user script)

DMTCP & CRIU Plugin implementation:

Implementation (3): Modifications to Slurm

- Usual stuff related to a new plugin:
Makefiles...
- Checkpoint.h: include new libraries
- Spank:
 - New parameter **S_JOB_CHECKPOINT_DIR** so **spank_get_item** returns checkpoint folder
 - Creation of **spank_set_item** to modify parameters of the job.
 - At this moment, only “S_JOB_ARGV”

Current status

- CRIU Plugin
 - Finished & tested

- DMTCP Plugin
 - Main implementation already finished
 - Some issues being debugged in DMTCP related to Infiniband driver
 - When everything is working OK, it will be put on Slurm mailing list

BLCR Vs. CRIU Vs. DMTCP

- Disclaimer: this is our personal experience

	CRIU	DMTCP	BLCR
Integration with Slurm	YES	YES	YES
Requires application modification	NO	NO	Recompile app
MPI applications	NO	YES	YES
Can checkpoint running application without preloading	YES	NO	YES* library must be loaded
Overhead besides checkpoint	NONE	Init: sec. CPU: 1-2%	CPU:1-2%
Can checkpoint containers (Docker & LXD)?	YES* we have only tested Docker, not LXD	NO	NO
Infiniband support	N/A	YES	NO* we haven't tried, comes from doc.

Take a look!

<https://github.com/supermanue/slurm>



Future: short term

- Full comparison on checkpoint mechanisms (characteristics, performance, scalability...)
- Finishing implementation of a multi-checkpoint plugin that automatically determines which mechanism to employ on each job
- Both will be ready by the end of the year

Future: short term multi-checkpoint plugin algorithm

- Is the code integrated with BLCR?
 - Yes-> use BLCR
 - No
 - Is it parallel?
 - Yes -> Use DMTCP
 - No -> Use CRIU
- This is our current naive approach, will be modified after performance & scalability analysis

Future: middle and long term

- After Docker itself is well integrated in Slurm (security and so), integrate Docker checkpoint in Slurm.
- Employ checkpoint/restart for live job migration inside clusters

Thanks for your attention

manuel.rodriguez@ciemat.es