RAMSES SLURM DATABASE

Draft document, Pierre OCVIRK, 19/09/2024

CONTEXT

The purpose of this document is to help organise the RAMSES Slurm DB (RSDB) working group.
The RSDB is an attempt at improving the user experience when deploying ramses on computing centres. Depending on the queue system, the heterogeneity of the compute resources, the quality of documentation and the complexity of the simulation at hand and its management of data, generating / adapting a generic slurm script as provided by the HPC center documentation can be time-consuming and frustrating, in particular for new users. The RSDB WG aims at creating a database of such scripts for users to support deployment of RAMSES and speed up progress towards production.

STRUCTURE / CONTENT / METADATA

The main structure of the DB is not settled yet but a number of principles and requirements have been formulated:

- The slurm scripts ingested in the DB will be sourced from the community
- Each script should be provided with metadata allowing quick and efficient understanding of the compute profile and context. A metadata template will be provided to the community.
- The DB should be easily accessible, both for uploading and viewing / downloading scripts.
    - Upload is here:https://seafile.unistra.fr/d/7cf0c55d4a0447a8aba5/
    - Test db could be here: https://seafile.unistra.fr/d/3d5a5d9d8f764bfebd62/
    - Official DB lives on github for easy cloning
    - => in https://github.com/ramses-organisation/ramses-job-scripts/
-
- The DB is curated by RSDB steward(s). The role of these stewards is several fold:
    - Retrieve the user-uploaded scripts from a TBD repo, on a regular basis (once per week / every other week / once a month)
    - => AT RUM2025
    - Review the scripts to verify / assess provenance and conformity of metadata, if necessary interact with the author to clarify.
    - Ingest into the main DB.
    - There is a seafile repo for upload from users, and a github repo for download to users. The steward makes the link between the two.

- Data model: several metadata items have been identified and are presented in the table below:

Item	keyword	Type, format	mandatory/desirable/optional
Computing center name main acronym	cc_main_name	String, “”	mandatory
Computing center developed acronym	cc_main_full	String, “”	desirable
Computing center name alt acronym	cc_alt_name	String, “”	optional
Computing center developed alt acronym	cc_alt_full	String, “”	optional
Supercomputer name main	sc_main_name	String, “”	mandatory
Supercomputer name alt	sc_alt_name	String, “”	optional
Execution date	exec_date	String, “dd-mm-yyyy”	desirable
Simulation project name main (e.g. short)	proj_main_name	String, “”	desirable
Simulation project name alt (e.g. long)	proj_alt_name	String, “”	desirable
Principal Investigator / author / uploader	pi_name	String, “”	mandatory
Principal Investigator email	pi_email	String, “”	optional
associated publication bibcode	sim_bibcode	String, “”	desirable
Queue management system (slurm / sbatch / bsub / other)	sim_queue	String, “”	mandatory
NNODES	sim_nnodes	int	mandatory
NMPI domains	sim_nmpi	int	mandatory
NCPU	sim_ncpu	int	mandatory
NGPU	sim_ngpu	int	Optional (0) if no GPU
NTHREADS_TOTAL	sim_nthreads_total	int	mandatory
NVECTOR	sim_nvector	int	optional
Cpu Compiler: pgi/intel/gnu/cray	sim_cpu_compiler	String, “”	mandatory
Accelerator compiler: nvidia/amd/intel	sim_accel_compiler	String, “”	optional
Modules:	sim_modules	Comma-separated list,”,”	desirable

Original Filename	ori_file_name	String, “”	desirable
Original filename is the filename on the computing center, as it was executed.			
Alt filename	alt_file_name	String, “”	Leave blank / Workflow-assigned
Alternative filename, for conveniency, assigned by the workflow / curator. This should be the filename in the INCOMING dir, if different from the original filename.			
Database filename	db_file_name	String, “”	Leave blank / Workflow-assigned
Final name in the static database			
Static DB path	db_path	String, “”	Leave blank / Workflow-assigned

Example slurm script with metadata, proposed format:

RAMSES SLURM DB

```
########
#HEADER
########
ori_file_name=run_ramsesaton_summit_nobb_4096.lsf
alt_file_name=
db_file_name=
db_path=
cc_main_name=OLCF
cc_main_full=Oak Ridge Leadership Computing Facility
cc_alt_name=
cc_alt_full=
sc_main_name=Summit
sc_alt_name=
exec_date=21-01-2021
proj_main_name=Cosmic Dawn III
proj_alt_name=CoDaIII
pi_name=Pierre OCVIRK
pi_email=
sim_bibcode=2022MNRAS.516.3389L
sim_queue=slurm
sim_nnodes=4096
sim_nmpi=131072
sim_ncpu=131072
sim_ngpu=24576
sim_nthreads_total=131072
sim_nvector=32
sim_cpu_compiler=pgi
sim_accel_compiler=nvidia
modules=pgi/20.1,cuda/9.2.148



#######
#SCRIPT
#######
#! /bin/bash
# begin LSF directives
```

```
#BSUB -P AST031
#BSUB -W 24:00
#BSUB -nnodes 4096
#BSUB -alloc_flags gpumps
#BSUB -J 8192_131k
#BSUB -o 8192_131ko.%J
#BSUB -e 8192_131ke.%J
#BSUB -q batch
#BSUB -B
#BSUB -N

module load pgi/20.1
module load cuda/9.2.148
cd /gpfs/alpine/ast031/proj-shared/pocvirk/CoDaIII/prod_sr/
module list

# set last snap as restart
./set_last_snap_as_restart_sr
# set ramses log file name using date
now=$(date +"%Y-%m-%d-%Hh%Mm%Ss")
printf "%s\n" $now
rlogname="ramseslog-${now}"
echo "ramseslogname = $rlogname"

# Cosmic Dawn III 4096 nodes run
jsrun --nrs 8192 --tasks_per_rs 16 --cpu_per_rs 16 --gpu_per_rs 3 --rs_per_host 2 --latency_priority GPU-CPU
stdbuf -o0
./ramses_aton_128x128x256_noHe_noZ_DS_BPASS_6DIGITS_LONGINT_IOGS-R=32-8192_ULM=4_NOMKDI
R_PARTSP2 ramses.nml > $rlogname
echo "finished"
```

INDEXATION and repository structure

1 - STATIC_DB

The static DB will be a repository with the following structure:

- Computing Center
  - Computer
    - Execution year
      - Compiler

This structure is useful for a web-based exploration of the DB in a browser.

2 - Dynamic, Python-queriable DB

The full DB is anticipated to be small enough to be wholly downloadable by users in one tarball. A registry file will contain the metadata of all the slurm files, in a python dictionary structure. Using this file it should be easy to make queries such as:

- What is the most recent slurm executed on Jean Zay?
- Show me the 3 largest runs in terms of MPI parallelism
- Show me the SPHINX production run slurm script
- …

The structure for this will be a github repo containing the DB data and a few python scripts to read the metadata registry.

MISCELLANEOUS NOTES

- Should we add metadata describing the physics of the simulation? no
- What elements from simDM should we think about integrating? (https://arxiv.org/pdf/1402.4744)