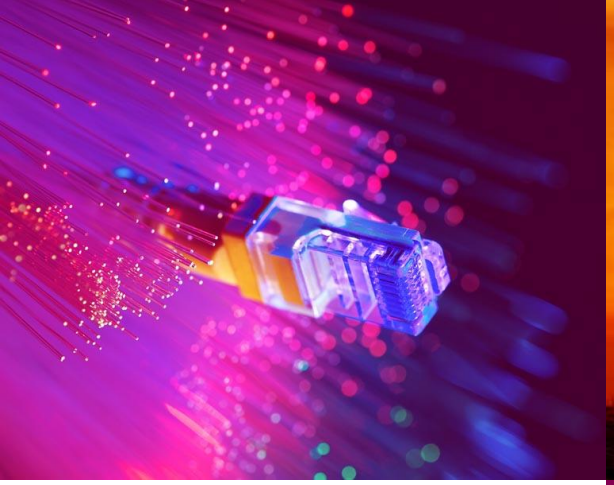
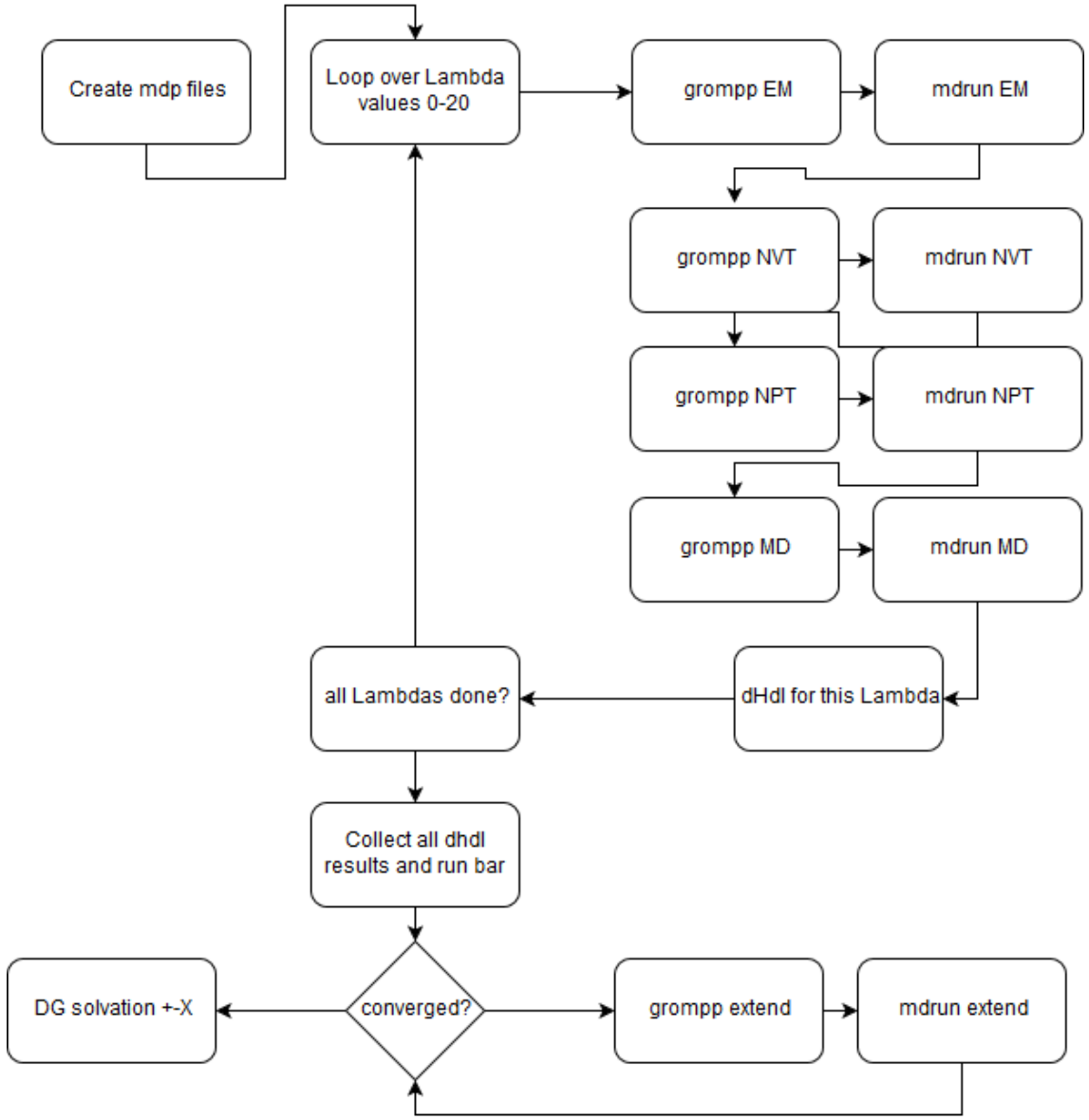


Workflows; some examples + NOMAD

Spring School on Computational Chemistry 2019 - Atte Sillanpää

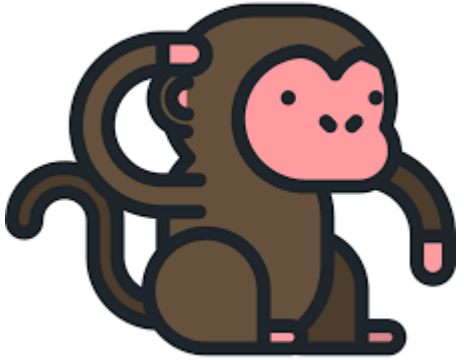
There are many ways to get the job done

- Revisit Gromacs hands-on from Tuesday (Thermodynamic Integration to get methane solvation free energy ΔG_{solv})
  - Run 21 sets of simulations
  - Run an analysis program
  - Check for convergence
  - Extend if needed



The monkey-work way

- Run each step manually changing lambda for each job
- Slow
- Boring
- Error prone
- Do this once, and when it works, automate
- Have coffee



Bash script - locally or as batch jobs

- Put the commands into a script
- And run it
  - Loops will remove the monkey-work
- Next job runs when previous is done - takes long
- Instead of running locally - submit to a queue
  - →N batch jobs → your HPC center contact will raise eyebrows

Array jobs 1/2

- Based on similar bash scripts at core
- An array job is ”one” job with lots of subjobs
  - Your HPC center contact approves
  - Efficient use of resources
  - Straightforward syntax
  - Can control how many run at a time
  - ”Light” for the queueing system to manage
  - Easy for you to monitor
  - Likely preinstalled - all you need to do is use

Array jobs 2/2

- Best suited for running the a similar job many times
- **`#SBATCH --array=1-100:50`**
- Defines to run 100 jobs, where a variable **`$SLURM_ARRAY_TASK_ID`** gets each number (**`1,2,...100`**) in turn as its value. This is then used to launch the actual job e.g. :
- **`srun myprog input_$SLURM_ARRAY_TASK_ID > \`**
  `output_$SLURM_ARRAY_TASK_ID)`
- Thus this would run 100 jobs:

```
srun myprog input_1 > output_1
srun myprog input_2 > output_2
…
srun myprog input_100 > output_100
```

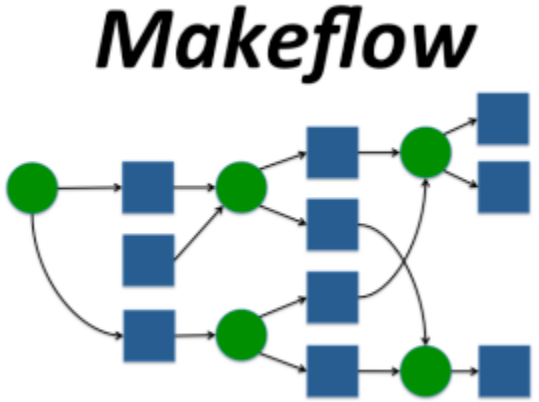
- For more information: research.csc.fi/taito-array-jobs

FireWorks workflow manager

- Made for submitting and managing large number of jobs
- Can make use of the same bash scripts (but does not have to)
- Facilitates monitoring (rerun failed jobs etc.)
- Takes some time to learn and script
  - But then repurposable
  - Shareable, documents your work
- YAML, Python, MongoDB
- Interfaces to many resources

Makeflow workflow manager

- A simple solution
- Easy to set up
- Based on the Makefile syntax
- Some limitations
  - All jobs with the same resources
  - Subset of Makefile syntax available
  - No loops
- Added value wrt. bash scripts?
  - Dependencies created automatically
  - Interface to multiple resources (Slurm, local, Condor, …)




```
target … : prerequisites …
        command
        …

target2 … : prerequisites2 …
        command
        …
```

Makeflow example - part of basic MD hands-on from Tuesday

```
CORES = 4
em2.gro: em2.tpr
	gmx mdrun -nt 4 -deffnm em2

nvt.tpr: em2.gro nvt.mdp
	gmx grompp -c em2.gro -r em2.gro -f nvt.mdp -o nvt.tpr

nvt.gro nvt.edr nvt.log: nvt.tpr
	gmx mdrun -nt 4 -deffnm nvt

npt.tpr: nvt.gro npt.mdp
	gmx grompp -c nvt.gro -r nvt.gro -f npt.mdp -o npt.tpr

npt.gro npt.edr npt.log: npt.tpr
	gmx mdrun -nt 4 -deffnm npt
```

```
target ... : prerequisites ...
	command
	...

target2 ... : prerequisites2 ...
	command
	...
```

Disclaimer…

- My example does not conform to all rules
  - Just one input and last output (which creates a dependency later) listed
  - Would fail if files should be sent elsewhere rather than used locally
- Conditional execution?
  - Couldn't figure out how to continue or not after convergence check
- No loops in DAG, no loops in Makeflow
  - Loop outside (while creating your makeflow)

Another approach - No fancy tools

- Learn to use one tool really well, and use it for all
- Same job.sh, same extend.sh same first array.sh
- Submit runbar with dependency to first array.sh
- It submits array-extend for unconverged lambda + another copy of itself with dependency to array-extend
- → once no longer lambdas need extension, done
- https://github.com/jlento/atm-doc/tree/master/workflows
- Think before you "buy", support close by is valuable

NOMAD - Novel Materials Discovery Laboratory

- Repository for your computational results
- Just upload - all metadata added automatically
- Conversion of data to uniform format
- Discoverable from Materials Encyclopedia
- Queryable from Analytics Notebooks
- 3 minute introductory video: https://youtu.be/yawM2ThVlGw
- https://www.nomad-coe.eu/