

The BBH Factory: Herding Simulations

Erik Schnetter
Baton Rouge, October 2007





Overview

- “BBH factory” is a glorified set of scripts to:
 - maintain source code on different machines
 - build executables on difficult machines
 - manage many simulations or many machines
- Peter likes it!



Outline

- Introducing some concepts
- Examples for copying source code, building executables, managing simulations
- Future possibilities
- Maybe an interactive demo



Source Code on Different Machines

- Problem: When developing new code, we don't want to use CVS before it has seen some testing
- Unfortunately, we have to use many machines (laptop, Catbert, Eric, Abe, etc)
- Copying source code manually is confusing



Source Code on Different Machines

- Idea: Use one machine (e.g. laptop) as “master”, all other machines should have identical copies
- Use rsync to update the other machines
- Handle multiple versions of source trees
- Details: speed, access control, hidden machines, directory names, disk space, ...



Source Code on Different Machines

- `bbhfactory/bbh rsync eric`
- `bbhfactory/bbh remote eric rsync abe`
- `bbhfactory/bbh help`



Building Executables

- Problem: Each machine requires different compilers, compiler options, uses different MPI versions, has libraries installed in different places, etc
- Autoconf does not work



Building Executables

- Idea: Store options and library names/locations for every machine
- Determine automatically on what machine one compiles
- Remember a qsub script file which fits MPI version of executable
- Details: Find out machine name, remember option lists, make commands, etc



Building Executables

- `bbhfactory/bbh remote eric`
`build einstein-eric-mvapich-debug`
`--optionlist einstein-eric-mvapich-debug`
`--thornlist einstein-eric`
`--scriptfile eric-mvapich`
- `bbhfactory/bbh list-configurations`



Managing Simulations

- Problem: Queue times are short, machines are unreliable, people make mistakes
- Problem: Parameter/resolution/scaling studies require many runs
- Problem: Accidentally writing output to the wrong directory is disastrous



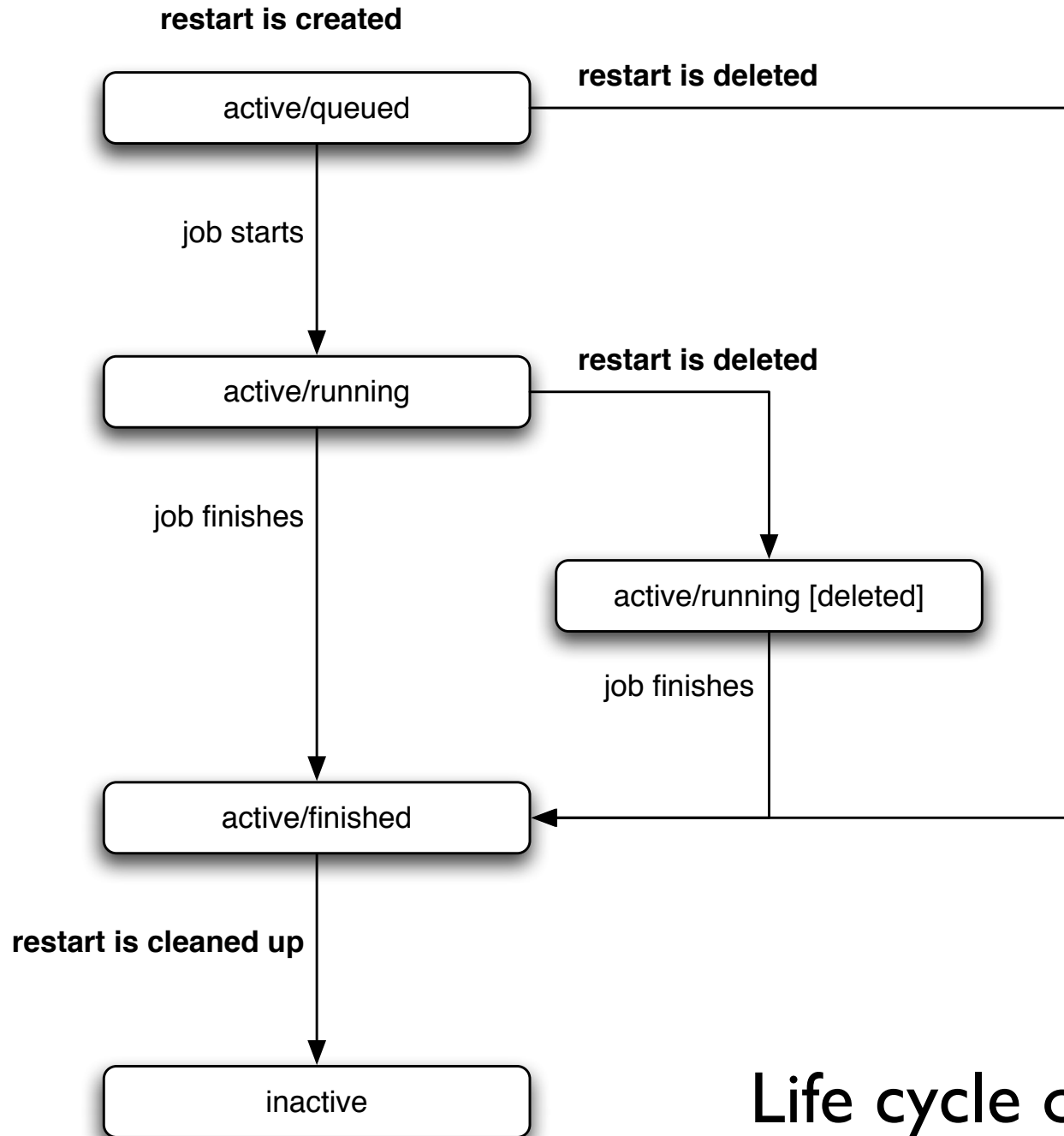
Managing Simulations

- Some concepts:
 - There are *simulations* and *restarts*
 - Each simulation can contain many restarts
 - The simulation defines executable and parameter file, i.e., physics and numerics
 - The restart defines only processor count, queue wall time, etc



Managing Simulations

- To create a simulation:
 - `bbhfactory/bbh create qc0-reference-0000`
`--configuration einstein-eric-mvapich-optimise`
`--parfile parfiles/einstein/qc0-reference.par`
- To submit a simulation:
 - `bbhfactory/bbh submit qc0-reference-0000`
`--procs 48`
 - `bbhfactory/bbh restart qc0-reference-0000`
`--procs 480 --walltime 24:00:00`



Life cycle of a restart



Managing Simulations

- Restarts need to be *cleaned up* before they can be resubmitted again
- This marks the simulation as inactive, preventing double-submitting
- Cleaning up also removes half-written output files, corrects file permissions, etc



Managing Simulations

- Other commands:
 - `bbhfactory/bbh list-simulations`
 - `bbhfactory/bbh remote eric status qc0-reference-0000 --verbose`
 - `bbhfactory/bbh remote eric show-output qc0-reference-0000`



Templates

- Script files and parameter files are templates
- Certain variables are automatically replaced
- Variable references look like `@VARIABLE@`
- Pre-defined variables contain the host name, directory name, number of nodes, number of processors, wall time, etc
- Additional variables can be defined with `--define SOMEVAR=SomeValue`



Additional Features

- Executables and checkpoint files are not copied but hardlinked, saving space
- There is a *machine database* (a perl hash table) describing the peculiarities of all known machines



Conclusion

- The BBH factory simplifies maintaining source trees on different machines, building executables, and managing simulations
- There is no magic in there, each feature is “simple”
- Nevertheless it’s quite useful when herding simulations
- Could be extended for non-Cactus simulations
- A graphical user interface could be added (Parca)