Workflows with nextflow

Brett Milash

Center for High Performance Computing

brett.milash@utah.edu

nextflow

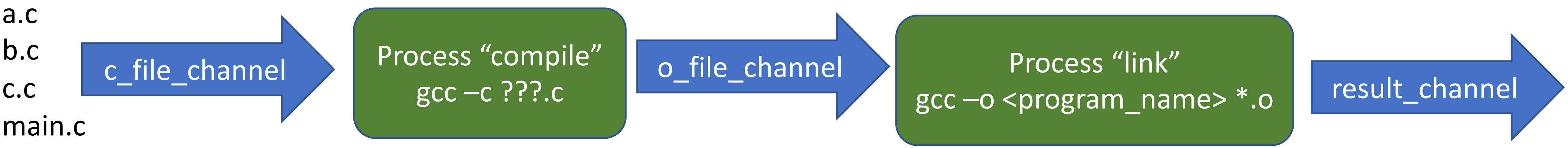
- Workflow manager written in Java
- Based on the dataflow programming model
  - https://en.wikipedia.org/wiki/Dataflow_programming
- Available at CHPC:
  - “module load nextflow”
- Install your own:
  - “curl -s https://get.nextflow.io | bash; chmod +x ./nextflow”
- Documented here: https://www.nextflow.io/

nextflow Workflows

Workflows consist of

- processes – these do the work
- channels – these carry data between processes
- executor – specifies where work gets done (local, cluster, cloud, etc.)

Written in Groovy (https://groovy-lang.org/), an interpreted Java



a.c
b.c
c.c
main.c

c_file_channel

Process “compile”
gcc –c ???.c

o_file_channel

Process “link”
gcc –o <program_name> *.o

result_channel

nextflow Processes

- Processes have names
- Can receive input from one or more channels
- Can send output to one or more channels
- Have a script section written in any scripting language
  - default is bash
  - use “#!” notation to specify a different language, e.g. “#!/usr/bin/env perl”
  - caveat: languages besides bash not fully integrated
- Get executed in a directory of their own
  - defaults to “./work/xx/yyyyy”
- Can “publish” results back to some specified directory (not by default)

nextflow Channels

- Have names
- Any type of data: strings, numbers, files, etc.
- Emit 1 value at a time (by default)
- Can be modified by Operators
    - collect (all the values and emit at one time)
    - collate (values into groups)
    - filter
    - find unique values
    - find distinct values
    - and many many others

nextflow Executors

- Provide abstraction of where the processes are executed
- Default is local execution
- Other options are SLURM, Open Science Grid, AWS, Google Cloud
- Specify executor for entire workflow (in nextflow.config), and can override on a per-process basis (in the workflow itself)
- Made possible by “./work/xx/yyyyy” process directory

```
process {
    executor='slurm'
    queue='notchpeak-shared'
    account='my_slurm_account'
    cpus = 1
    memory = 16.0G
}
```

```
process {
    executor=‘local’
}
```

nextflow.config file

- Optional file provides a global configuration for your workflow
- Includes executor, caching, publishing directory defaults for all processes
- Can be overridden with directives in each process

```
process {
    executor='slurm'
    queue='notchpeak-shared'
    account='my_slurm_account'
    cpus = 1
    cache=true
}
```

nextflow.config file

```
process link {
      executor local
      cache false
      module 'gcc/8.3.0’
      publishDir '.’
      input:
```

workflow file

Results caching

- nextflow can behave like “make” or “snakemake”, where only the out-of-date targets are computed
- This is not the default behavior
- To enable this:
    - Set “cache = true” in nextflow.config
    - Run with “-resume” flag, e.g. “nextflow run my_workflow.nf –resume”
- When enabled, changes to input or process script trigger re-execution of a process

What makes nextflow special?

- The workflow itself is one of the dependencies (when caching)
  - A change to a process triggers re-execution of that process (and its dependents)
- Executor is separate from the workflow
  - Changing where execution happens is separate from the workflow definition
- Channels are not just for files
  - Can carry any type of value (e.g. numbers, strings, files, including file pairs)
  - Behavior an be modified using Operators
- Any scripting language can be used in a process (with some caveats)

Thank you for coming!

- Questions or comments are welcome!
- brett.milash@utah.edu