

Familiar?



Today, I needed to run one more experiment because I forgot some information on how I got the nice results I want to publish and my supervisor wants to check everything. So I copied the data again on Lemaitre2 and submitted the 5 jobs. I then realized I copied the wrong input file and had to cancel all jobs and resubmit them. But then I forgot to redefine the dependencies so one started too early – the intermediate result wasn't ready yet – and I have to resubmit it. And of course now Lemaitre2 is full so I should probably start everything again on NIC4. I really hate my



Consortium des Equipements
de Calcul Intensif
en Fédération Wallonie-Bruxelles

Introduction to Scientific Workflow Management

david.waroquiers@uclouvain.be
damien.francois@uclouvain.be
October 2015



You follow workflows...

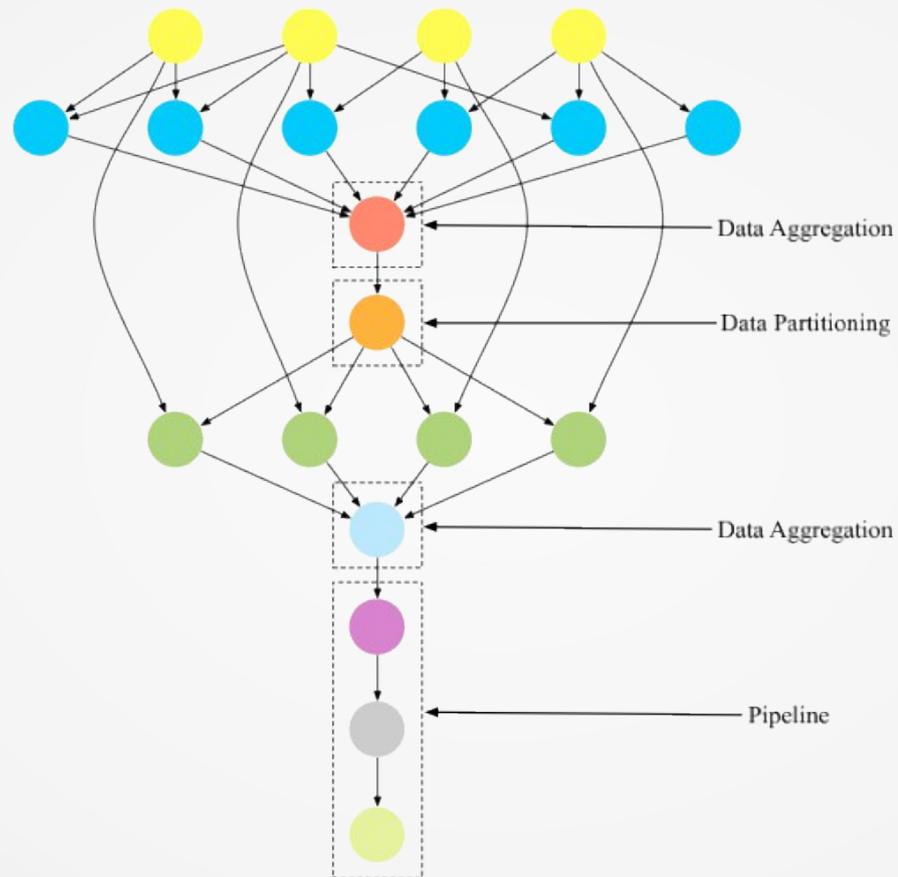


“A workflow is a precise description of a scientific procedure — a multi-step process to coordinate multiple tasks, acting like a sophisticated script”

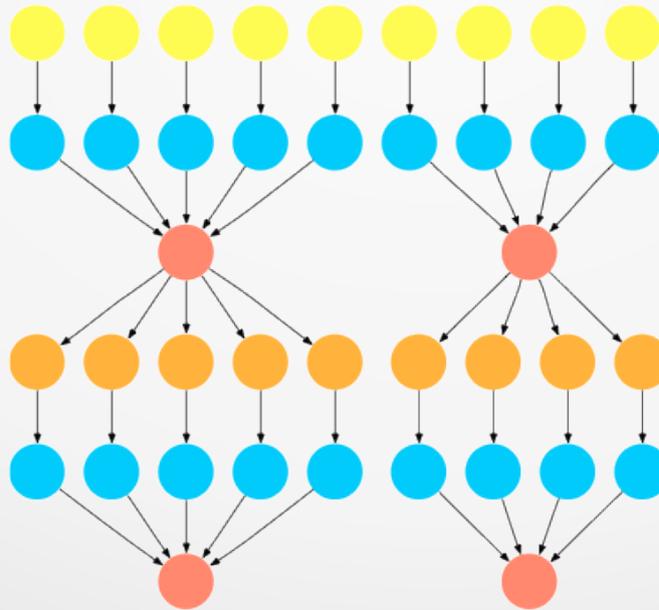
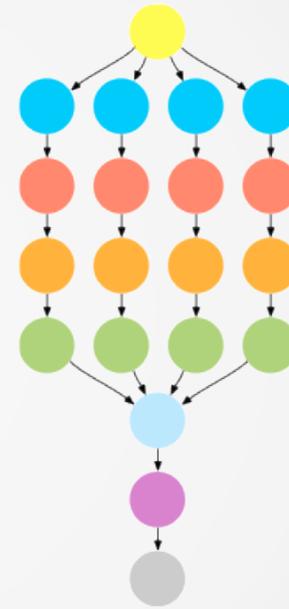
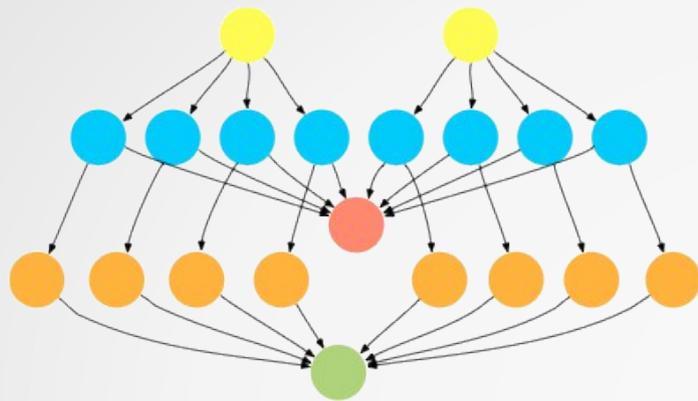
Yours can be simple..



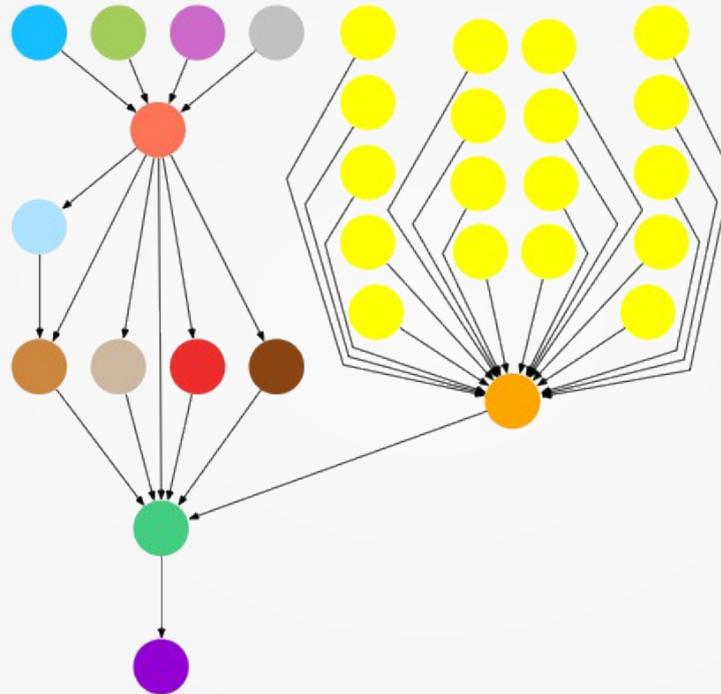
.. or complex



.. or complex



.. or even bizarre



By hand?



- Error prone
- Cumbersome
- Does not scale

- How do you share?
- How do you explain?
- How do you recall provenance?

Workflow systems



Platforms that offer

- (1) invocation of the service applications and handling the heterogeneity of data types and interfaces on multiple computing platforms;
- (2) monitoring and recovery from failures;
- (3) optimization of memory, storage, and execution, including concurrency and parallelization;
- (4) data handling: mapping, referencing, movement, streaming, and staging;
- (5) logging of processes and data provenance tracking; and
- (6) security and monitoring of access policies.

Using a workflow system?



- Error prone
- Cumbersome
- Does not scale

No as it is automated



- How do you share?
- How do you explain?
- How do you recall provenance?

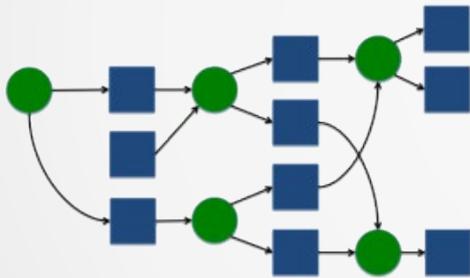
All in one file



Makeflow = Make + Workflow



Makeflow



```
Sample Makeflow to process each file with
# lower.sh then upper.sh
#
d1.tmp: d1.txt lower.sh
        ./lower.sh d1.txt d1.tmp

d2.tmp: d2.txt lower.sh
        ./lower.sh d2.txt d2.tmp

d3.tmp: d3.txt lower.sh
        ./lower.sh d3.txt d3.tmp

d4.tmp: d4.txt lower.sh
        ./lower.sh d4.txt d4.tmp

d1.res: d1.tmp upper.sh
        ./upper.sh d1.tmp d1.res

d2.res: d2.tmp upper.sh
        ./upper.sh d2.tmp d2.res

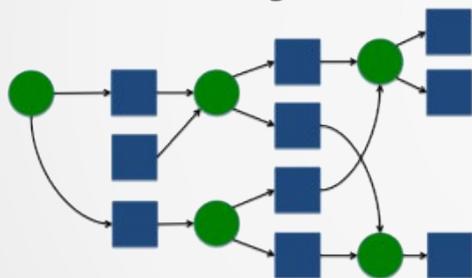
d3.res: d3.tmp upper.sh
        ./upper.sh d3.tmp d3.res

d4.res: d4.tmp upper.sh
        ./upper.sh d4.tmp d4.res
```

Makeflow = Make + Workflow

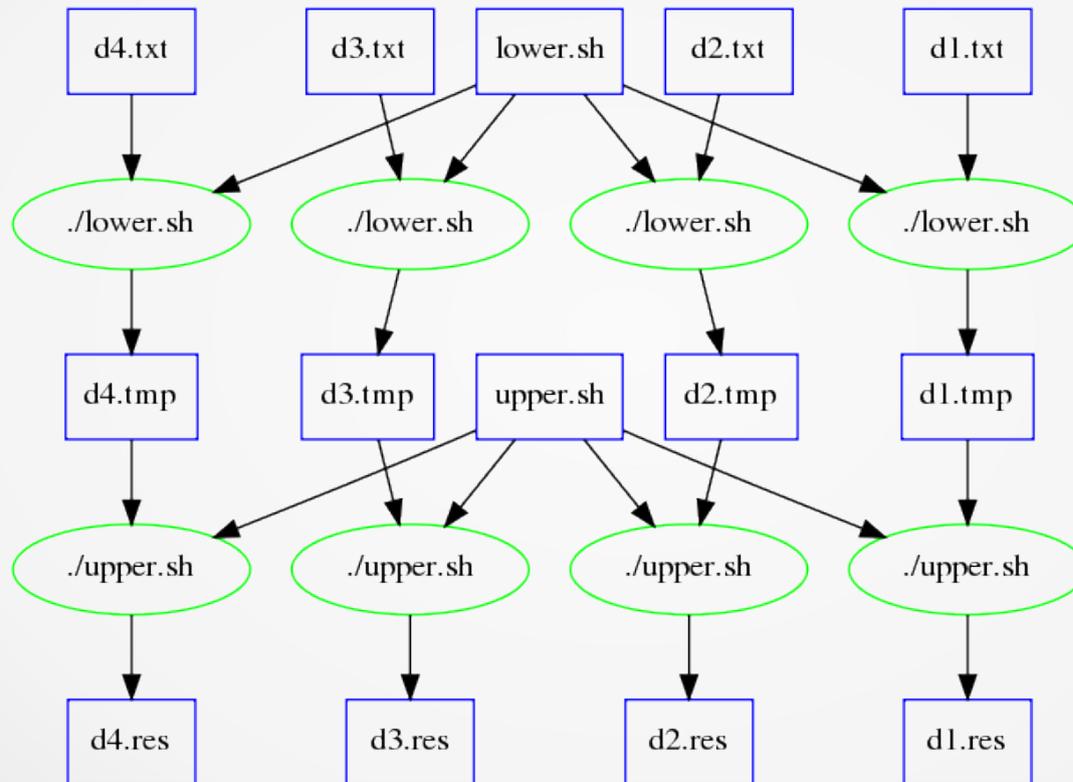


Makeflow



```
dfr@hmem00:~/parcomp $ makeflow -T slurm Makefile.expanded
parsing Makefile.expanded...
checking Makefile.expanded for consistency...
Makefile.expanded has 8 rules.
recovering from log file Makefile.expanded.makeflowlog...
starting workflow...
submitting job: ./lower.sh d4.txt d4.tmp
submitted job 643315
submitting job: ./lower.sh d3.txt d3.tmp
submitted job 643316
submitting job: ./lower.sh d2.txt d2.tmp
submitted job 643317
submitting job: ./lower.sh d1.txt d1.tmp
submitted job 643318
job 643315 completed
submitting job: ./upper.sh d4.tmp d4.res
submitted job 643319
job 643316 completed
submitting job: ./upper.sh d3.tmp d3.res
submitted job 643320
job 643317 completed
submitting job: ./upper.sh d2.tmp d2.res
submitted job 643321
job 643318 completed
submitting job: ./upper.sh d1.tmp d1.res
submitted job 643322
job 643319 completed
job 643320 completed
job 643321 completed
job 643322 completed
nothing left to do.
```

Dependency graph



Passing Slurm options



|| Batch Job Refinement

When executing jobs, Makeflow simply uses the default settings in your batch system. If you need to pass additional options, use the `BATCH_OPTIONS` variable or the `-B` option to Makeflow.

When executing jobs, Makeflow simply uses the default settings in your batch system. If you need to pass additional options, use the `BATCH_OPTIONS` variable or the `-B` option to Makeflow.

When using Condor, this string will be added to each submit file. For example, if you want to add `Requirements` and `Rank` lines to your Condor submit files, add this to your Makeflow:

```
BATCH_OPTIONS = Requirements = (Memory>1024)
```

When using SGE, the string will be added to the `qsub` options. For example, to specify that jobs should be submitted to the `devel` queue:

```
BATCH_OPTIONS = -q devel
```

Wrapper



```
dfr@hmem00 — bash
dfr@hmem00:~/parcomp $ makeflow --wrapper "/usr/bin/time -p" Makefile.expanded
parsing Makefile.expanded...
checking Makefile.expanded for consistency...
Makefile.expanded has 8 rules.
recovering from log file Makefile.expanded.makeflowlog...
starting workflow...
submitting job: /usr/bin/time -p ./upper.sh d4.tmp d4.res
submitted job 57341
submitting job: /usr/bin/time -p ./upper.sh d3.tmp d3.res
submitted job 57342
submitting job: /usr/bin/time -p ./upper.sh d2.tmp d2.res
submitted job 57343
submitting job: /usr/bin/time -p ./upper.sh d1.tmp d1.res
submitted job 57344
real 4.01
user 0.00
sys 0.00
job 57343 completed
real 4.01
user 0.00
sys 0.00
rreal 4.01
us4.01e
rus0.00e
rsy0.00s
s0.00y
s 0.00
job 57341 completed
job 57342 completed
job 57344 completed
nothing left to do.
dfr@hmem00:~/parcomp $
```

Other options



```
dfr@hmem00 ~$ bash
-f,--summary-log=<file>      Write summary of workflow to this file upon success or failure.
-j,--max-local=<#>          Max number of local jobs to run at once.      (default is # of cores)
-J,--max-remote=<#>        Max number of remote jobs to run at once.
                              (default 1000 for -Twq, 100 otherwise.)
-l,--makeflow-log=<logfile> Use this file for the makeflow log.          (default is X.makeflowlog)
-L,--batch-log=<logfile>   Use this file for the batch system log.     (default is X.<type>log)
-m,--email=<email>         Send summary of workflow to this email address upon success or failure.
-o,--debug-file=<file>    Send debugging to this file. (can also be :stderr, :stdout, :syslog, or :journal)
-R,--retry                 Automatically retry failed batch jobs up to 100 times.
-r,--retry-count=<n>      Automatically retry failed batch jobs up to n times.
  --wait-for-files-upto=<n> Wait for output files to be created upto n seconds (e.g., to deal with NFS semantics).
-S,--submission-timeout=<#> Time to retry failed batch job submission. (default is 3600s)
-X,--change-directory     Change directory: chdir to enable executing the Makefile in other directory.
-z,--zero-length-error    Force failure on zero-length output files

*Monitor Options:

-M,--monitor=<dir>        Enable the resource monitor, and write the monitor logs to <dir>.
  --monitor-limits=<file> Use <file> as value-pairs for resource limits.
  --monitor-interval=<#> Set monitor interval to <#> seconds.          (default is 1 second)
  --monitor-with-time-series Enable monitor time series.          (default is disabled)
  --monitor-with-opened-files Enable monitoring of opened files. (default is disabled)
  --monitor-log-fmt=<fmt> Format for monitor logs.          (default resource-rule-%06.6d)
```

Makeflow



- Syntax similar to make
- Able to submit jobs to Slurm
- Simple but useful

- No file input/output management
- Does not scale to hundreds of jobs

FireWorks A graphic of a firework exploding, with a central square and several lines radiating outwards to smaller squares, representing sparks or light trails.

“Give me six hours to chop down a tree and I will spend the first four sharpening the axe.”
- Abraham Lincoln

From now on...



Today, I had a walk by the lake and I had a great new idea about the parameter choice. When I came back to the office, I simply adapted the workflow and relaunched it just before attending the seminar .

I then left for lunch while my jobs were submitted and monitored by the workflow management system. I know that 2 jobs crashed because of a hardware issue and they were rescheduled so I will get the result a bit later. I hope the result will be spectacular. My lab mates wonder how I work so quickly.. I really should someday talk to them about Fireworks and Makeflow, etc. Anyway, I love my work since I attended