

Introduction to Workflow Management

and Workflow Manager Version 1.0

Christopher Harrop (Christopher.W.Harrop@noaa.gov)
NOAA/ESRL, GSD2
325 Broadway
Boulder, CO 80305-3337

Outline

- Introduction
- Common ad-hoc approaches
- Ensemble workflow design approaches
- Workflow design tips
- Existing workflow management engines
- Workflow Manager 1.0 how-to

Introduction

Introduction

- A workflow is a collection of interconnected steps employed to accomplish an overall goal
- Workflow management systems provide
 - A means of *defining* a workflow
 - *Automation* of workflow execution

Introduction

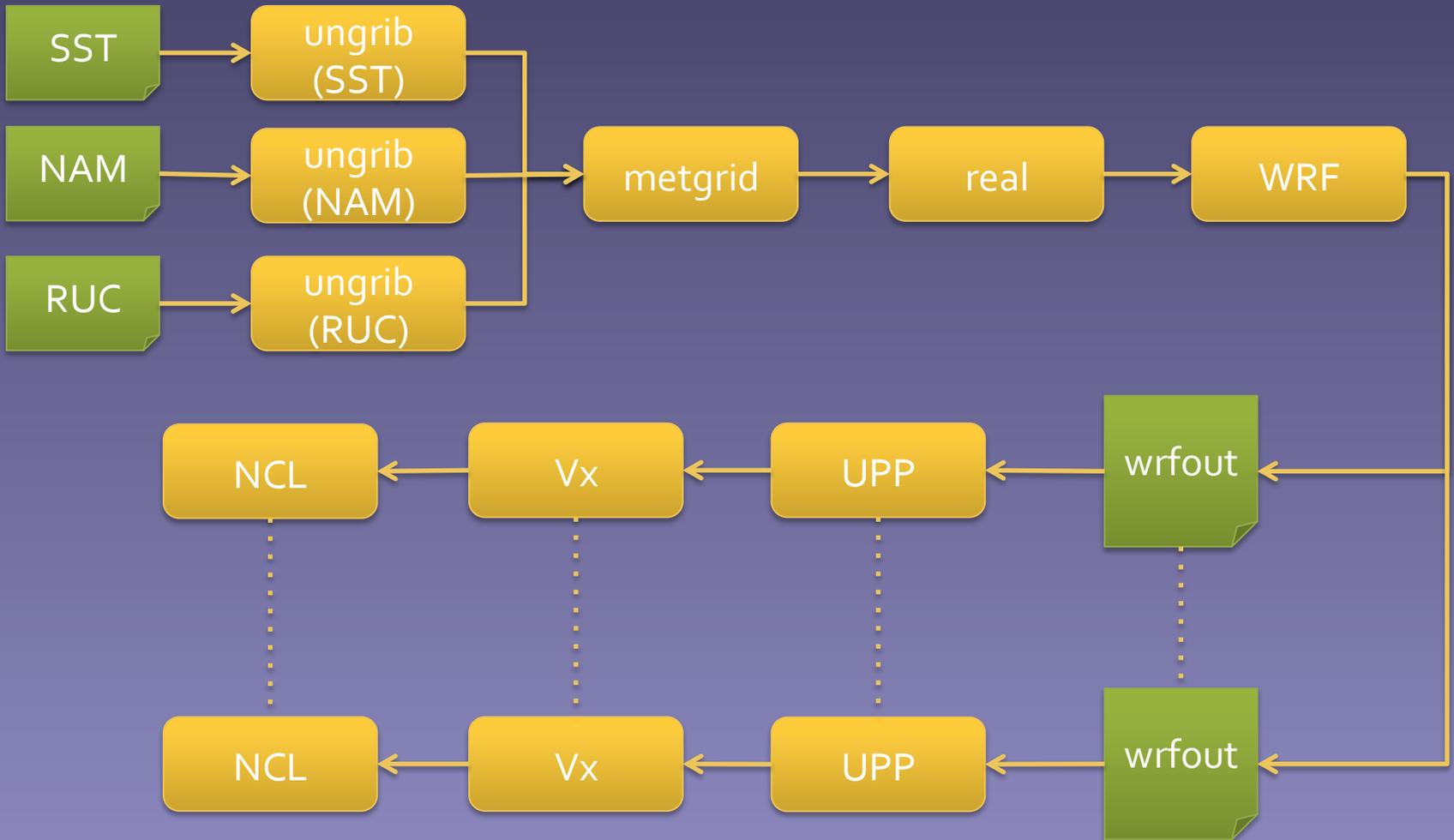
- Why formalized workflows?
 - Concrete representation of complex computations
 - Reuse
 - Facilitates provenance

Introduction

- Why use a workflow management system?
 - Complexity
 - Scale
 - Reliability
 - Efficiency
 - Provenance

Common Ad-hoc Approaches

Common Ad-hoc Approaches



Common Ad-hoc Approaches

- The “driver” script
 - Relies on long uptime
 - State embedded in script execution
 - No fault tolerance
 - Manual fault recovery
 - Not feasible for complex workflows
 - Not reusable

```
#!/bin/sh

# Run step one
qsub ./step_one.sh

# Run step two
qsub ./step_two.sh

# Run step three
qsub ./step_three.sh
```

Common Ad-hoc Approaches

- The “linear job chain”
 - No concurrency
 - Relies on unbroken job chain
 - No fault tolerance
 - Manual fault recovery
 - Can lose control of workflow execution
 - Not reusable

The script for step N

```
#!/bin/sh

# Execute step N
./run_step_N.exe

# Submit step N+1
qsub ./step_N_plus_1.sh
```

To start the job chain

```
qsub ./step_1.sh
```

Common Ad-hoc Approaches

- The “tree job chain”
 - Allows concurrency
 - Relies on unbroken job chain
 - No fault tolerance
 - Manual fault recovery
 - Can lose control of workflow execution
 - Not reusable

The script for step N

```
#!/bin/sh
```

```
# Execute step N  
./run_step_N.exe
```

```
# Submit steps N+1 N+2  
qsub ./step_N_plus_1.sh  
qsub ./step_N_plus_2.sh
```

To start the job chain

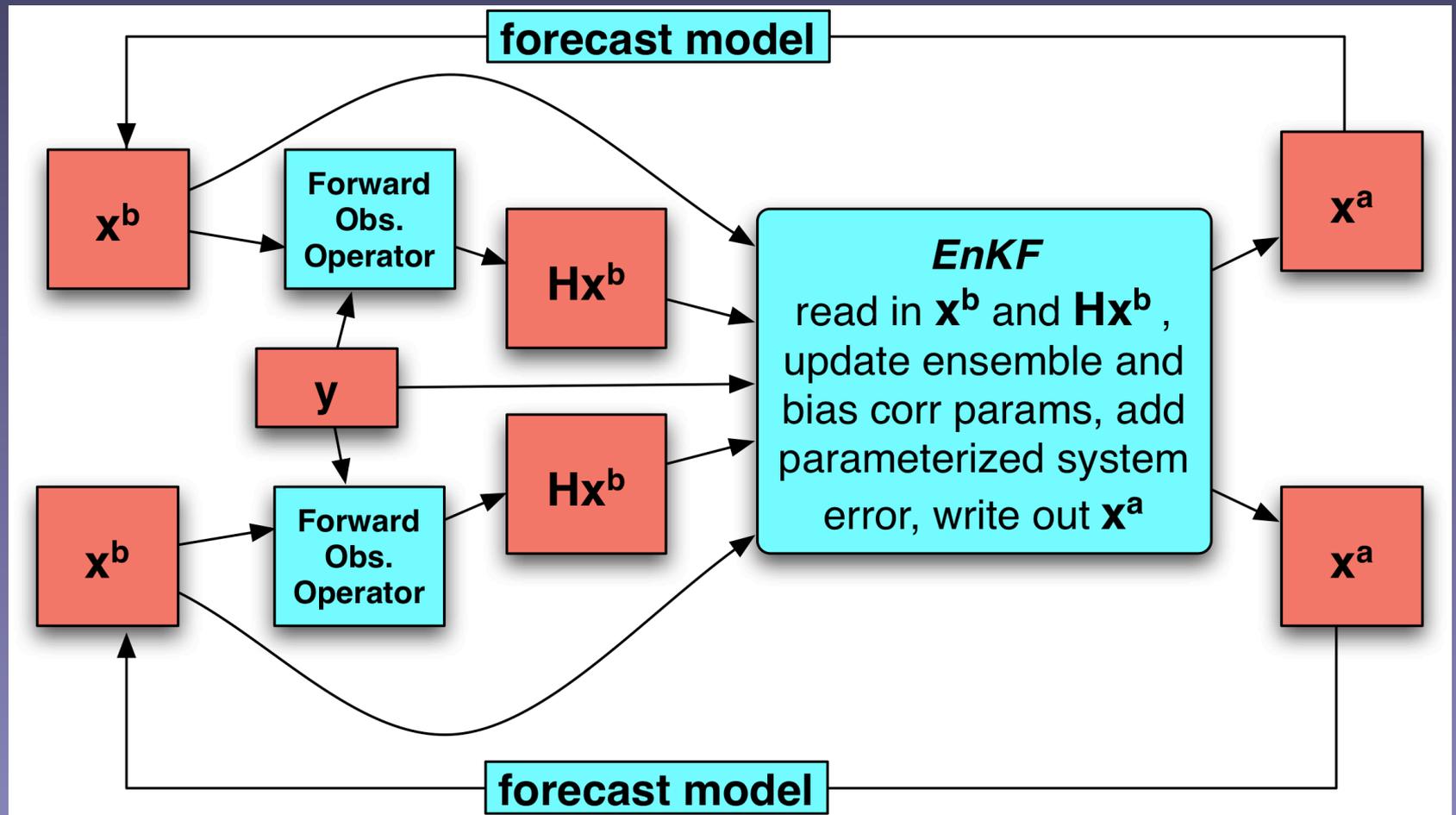
```
qsub ./step_root.sh
```

Common Ad-hoc Approaches

- Some deficiencies can be addressed by
 - Recording workflow state on disk
 - Retrying job submission failures
 - Careful error checking
- Robust fixes for most deficiencies is hard

Ensemble Workflow Design

Ensemble Workflow Design



Ensemble Workflow Design

- Are ensemble workflows fundamentally different?
- Two approaches
 - Each of the M workflow steps runs N copies of the executable concurrently (one batch job for all members per step)
 - Each of the M workflow steps is duplicated N times, (one batch job per step per member)

Ensemble Workflow Design

- Run all N members concurrently as one step (one batch job for all members)
 - Requires scheduling a job N times as large
 - Must manage all failures of individual members with ad-hoc methods
 - Rerunning only the failed members is complicated or impossible
 - Increased probability of a failure
 - Not portable
 - Requires detailed knowledge of hostfile management and machine architecture

Ensemble Workflow Design

- Run N members concurrently as N independent steps (one batch job per member)
 - Requires scheduling N small jobs instead of one large one
 - No complex, unreliable, scripting logic to handle failures
 - Reruns of failed members is simple and efficient
 - Lower probability of failure
 - Portable
 - No hostfile manipulation or architecture knowledge required

Workflow Design Tips

Workflow Design Tips

- Create *atomic, autonomous* components
 - Building large things from small things is easy and reliable
- Isolate model components from automation systems
 - Allows model to be run by different automation systems
- Cluster tasks together when appropriate
 - Increases workflow efficiency
 - Short, small, tasks go well together
 - Small tasks do not go well with large tasks unless the small tasks are very short

Existing Workflow Management Systems

Existing Workflow Management Systems

- Many scientific workflow engines exist
 - Kepler, Pegasus, Taverna, Triana, ...and more
- What's wrong with them?
 - Nothing, but...impractical for our requirements
 - Often complicated to install and use
 - Heavy emphasis on Grand Challenge applications
 - Service Oriented Architecture (SOA)

Workflow Manager 1.0

Workflow Manager 1.0

- Completely redesigned
 - Improved throttling
 - Improved cycle specification
 - Support for script arguments
 - Generic, portable, batch specifications
 - New dependency operators
 - Multi-threaded job submission
 - Improved fault tolerance
 - Improved logging and reporting

Workflow Manager 1.0

- Running the Workflow Manager

```
workflowmgr -d <database> -w <workflow>
```

- Must be run repeatedly to complete a workflow
- Each run has *potential* to advance workflow state
- Run interactively during setup and debugging
- Run from cron during production
- All scheduling is done by underlying resource manager

Workflow Manager 1.0

- Defining a workflow
 - Custom XML language
 - Use your editor of choice
 - Build templates and use XML generation tools
 - Specify steps
 - Script to submit
 - Dependencies
 - Runtime resource requirements
 - Environment

Workflow Manager 1.0

- The XML Header
 - Tells the parser what kind of document it is
 - Define ENTITIES for commonly used values

```
<?xml version="1.0"?>
<!DOCTYPE workflow
[
  <!ENTITY EXP_HOME "/the/path/to/my/experiment"
  <!ENTITY WRFV3 "&EXP_HOME;/WRFV3">
  <!ENTITY CYCLE_TIME "@Y@m@d@H">
]>
```

Workflow Manager 1.0

- The `<workflow>` tag
 - Specify resource manager
 - sge, moabtorque, lsf (easy to add more)
 - Specify run mode
 - Set throttling and expiration parameters

```
<workflow realtime="f" scheduler="sge"  
cyclelifespan="0:01:00:00" cyclethrottle="10"  
corethrottle="50" taskthrottle="50">
```

```
</workflow>
```

Workflow Manager 1.0

- The <cycledef> tag
 - Specify cycles to run
 - One instance of the entire workflow per cycle
 - cron-like specification
 - Minute, hour, day, month, year, day of week
 - Start, stop, step specification
 - Cycle groups

```
<cycledef group="group1">201101010000 201201010000  
1:00:00:00</cycledef>
```

```
<cycledef group="group1">0 */6 * * * *</cycledef>
```

Workflow Manager 1.0

- The <log> tag
 - Specify Workflow Manager logs
 - Usually want one log file per cycle
 - Specify logging verbosity

```
<log verbosity="0">/path/to/log/file</log>
```

Workflow Manager 1.0

- The `<cyclestr>` tag
 - Specify dynamic strings containing cycle date and time components
 - Specify offsets from current cycle
 - `<cycle_X/>` tags are no longer supported
 - Now uses `@` instead of `%` for time component flags

```
<cyclestr offset="2:00:00">test_@Y@m@d@H@M.log</cyclestr>
```

Workflow Manager 1.0

- The `<task>` tag
 - Specify what to run
 - Specify runtime requirements
 - Specify environment
 - Specify dependencies
 - Some attributes no longer supported

```
<task name="test" maxtries="3" cycledefs="group1">  
  
</task>
```

Workflow Manager 1.0

- The `<command>` tag
 - Specifies what to run
 - May use command line arguments

```
<task name="test">  
  
  <command><cyclestr>test.ksh -d @Y@m@d@H</cyclestr></command>  
  
</task>
```

Workflow Manager 1.0

- The `<envvar>` tag
 - Specifies environment settings to be passed
 - Values are optional

```
<task name="test">

  <envvar>
    <name>START_TIME</name>
    <value><cyclestr>@Y@m@d@H</cyclestr></value>
  </envvar>

</task>
```

Workflow Manager 1.0

- The runtime requirements tags
 - Specifies batch queue settings

```
<account>dtc</account>  
  
<jobname>test</jobname>  
  
<queue>hfip</queue>  
  
<cores>256</cores>  
  
<walltime>00:00:10</walltime>
```

Workflow Manager 1.0

- The runtime requirements tags (continued)
 - Specifies batch queue settings

```
<memory>512M</memory>
```

```
<join>/home/username/test/log/test.log</join>
```

```
<stdout>/home/username/test/log/test.out</stdout>
```

```
<stderr>/home/username/test/log/test.err</stderr>
```

```
<native>-ac flags=ADVRES:dtc-12z</native>
```

Workflow Manager 1.0

- The <dependency> tags
 - Specifies boolean expression of dependencies
 - Data dependencies
 - Task dependencies
 - Walltime dependencies
 - Dependency logical operators

```
<task name="test">  
  <dependency>  
  
  </dependency>  
</task>
```

Workflow Manager 1.0

- The <dependency> operators

Operator	True if....
<not>	The dependency is not satisfied
<and>	All dependencies are satisfied
<or>	At least one dependency is satisfied
<nand>	At least one dependency is not satisfied
<nor>	None of the dependencies are satisfied
<xor>	Exactly one of the dependencies is satisfied
<some>	The number of dependencies that are satisfied exceeds the given threshold

Workflow Manager 1.0

- The <dependency> operands
 - Task dependency

```
<taskdep task="taskname" cycle_offset="0" state="SUCCEEDED" />
```

- Data dependency

```
<datadep age="1:00">/path/to/file</datadep>
```

- Time dependency

```
<timedep>20120131000000</timedep>
```

Workflow Manager 1.0

- The `<dependency>` tag (continued)

```
<dependency>
  <or>
    <some threshold="0.5">
      <datadep>/path/to/data/input1.dat</datadep>
      <datadep>/path/to/data/input2.dat</datadep>
      <datadep>/path/to/data/input3.dat</datadep>
    </some>
    <and>
      <taskdep task="task1"/>
      <timedep><cyclestr>@Y@m@d@H@M@S</cyclestr></timedep>
    </and>
  </or>
</dependency>
```

Workflow Manager 1.0

- The `<metatask>` tag
 - Representation of large collections of similar tasks
 - Useful for ensemble workflows and post-processing
 - Can be nested without limit
 - Can contain multiple tasks
 - Can contain multiple `<var>` tags

```
<metatask>  
  
  <var id="member">01 02 03 04 05 06 07 08 09 10</var>  
  
  <task name="model_#member#">  
  </task>  
  
</metatask>
```

Workflow Manager 1.0

- Multi-threaded
 - workflowdbserver
 - workflowbqserver
 - workflowioserver
- Customization
 - .wfmrc
 - Turn server processes on/off (on by default)

Workflow Manager 1.0

- Tools
 - wfmstat
 - List current state of workflow
 - wfmrun
 - Force a task to run
 - wfmreport
 - Generate success rate report

Workflow Manager 1.0

- Debugging
 - Look at the workflow logs

```
Thu May 17 17:41:16 +0000 2012 :: fe5 :: Cannot submit test10, because maximum core throttle of 3 will be violated.
Thu May 17 17:41:16 +0000 2012 :: fe5 :: Submitted test1. Submission status is pending at druby://fe5:56185
Thu May 17 17:42:10 +0000 2012 :: fe5 :: Submission status of previously pending test1 is success, jobid=3543217
Thu May 17 17:42:12 +0000 2012 :: fe5 :: Task test1, jobid=3543217, in state QUEUED (qw)
Thu May 17 17:42:12 +0000 2012 :: fe5 :: Cannot submit test10, because maximum core throttle of 3 will be violated.
Thu May 17 17:43:13 +0000 2012 :: fe5 :: Task test1, jobid=3543217, in state SUCCEEDED (done), ran for 0.0 seconds,
exit status=0, try=1 (of 3)
```

Workflow Manager 1.0

- Debugging
 - Use the wfmstat utility

CYCLE	STATE	ACTIVATED	DEACTIVATED
200901010000	active	May 16 2012 18:08:04	-
200901021200	done	May 16 2012 18:08:04	May 16 2012 23:04:08
200901040000	active	May 16 2012 18:08:04	-
200901051200	active	May 16 2012 18:08:04	-
200901070000	active	May 16 2012 18:08:04	-
200901081200	active	May 16 2012 18:08:04	-
200901100000	active	May 16 2012 18:08:04	-
200901111200	active	May 16 2012 18:08:04	-
200901130000	done	May 16 2012 18:08:04	May 16 2012 23:04:08
200901141200	done	May 16 2012 18:08:04	May 16 2012 23:08:08
200901160000	done	May 16 2012 18:08:04	May 16 2012 23:04:08
200901171200	done	May 16 2012 18:08:04	May 16 2012 23:04:08
200901190000	active	May 16 2012 18:08:04	-
200901201200	active	May 16 2012 18:08:04	-
200901220000	done	May 16 2012 18:08:04	May 16 2012 23:04:08
200901231200	active	May 16 2012 18:08:04	-
200901250000	active	May 16 2012 18:08:04	-
200901261200	active	May 16 2012 18:08:04	-
200901280000	active	May 16 2012 18:08:04	-
200901291200	active	May 16 2012 18:08:04	-
200901310000	active	May 16 2012 23:08:05	-

Workflow Manager 1.0

- Debugging
 - Use the wfmstat utility

CYCLE	TASK	JOBID	STATE	EXIT STATUS	TRIES
200901010000	real_nmm	3488977	SUCCEEDED	0	1
200901010000	ungrib_NAM	3488676	SUCCEEDED	0	1
200901010000	wrf_nmm	3504048	DEAD	1	3
200901021200	real_nmm	3488980	SUCCEEDED	0	1
200901021200	ungrib_NAM	3488678	SUCCEEDED	0	1
200901021200	wrf_nmm	3489152	SUCCEEDED	0	1
200901040000	real_nmm	3488978	SUCCEEDED	0	1
200901040000	ungrib_NAM	3488679	SUCCEEDED	0	1
200901040000	wrf_nmm	3504049	DEAD	1	3
200901051200	real_nmm	3488979	SUCCEEDED	0	1
200901051200	ungrib_NAM	3488680	SUCCEEDED	0	1
200901051200	wrf_nmm	3504050	DEAD	1	3

Workflow Manager 1.0

- Documentation
 - <http://rdhpcs.noaa.gov/workflowmanager/>
 - Ongoing, needs more work
- Release
 - Soon, still cleaning up a few loose ends
 - Checkout from jetsvn repository with NEMS credentials

Workflow Manager 1.0

- Planned, but *unimplemented*
 - Metatask dependencies
 - Rollback (technically challenging)
 - Workflow Domain Specific Language (DSL)
 - Alternative to XML
 - Workstation workflows
 - Low priority

Workflow Manager 1.0

- Under *consideration*, but *unimplemented*
 - Automatic task clustering
 - Attempt to optimize throughput
 - Automatic run interval optimization
 - Colored Petri-Net (CPN) workflow model
 - More expressive than a DAG model

Questions

<http://rdhpcs.noaa.gov/workflowmanager/>