



# Nimrod projects

Colin Enticott  
Slavisa Garic  
Tom Peachey

Monash University



# Nimrod related projects

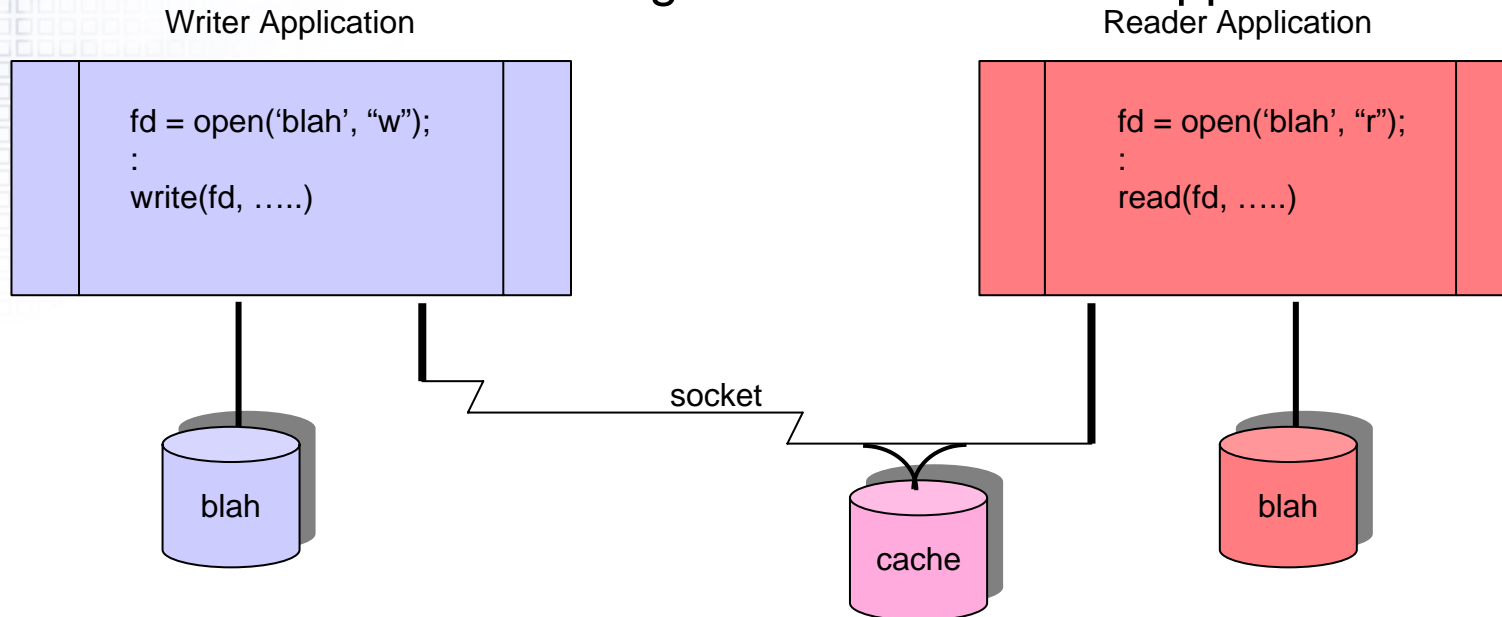
---

- GriddLes
- Nimrod with job dependencies
- Active Sheets
- Nimrod Java and WebServices API



# GriddLes

- <http://www.csse.monash.edu.au/~davidag/griddles>
- Legacy Applications are Components
  - Cannot necessarily modify them
- GriddLeS
  - Specification of the interconnections between components
  - Interfaces for discovering resources and mapping the computations to them
  - Locate data files in the grid and connect the applications to them



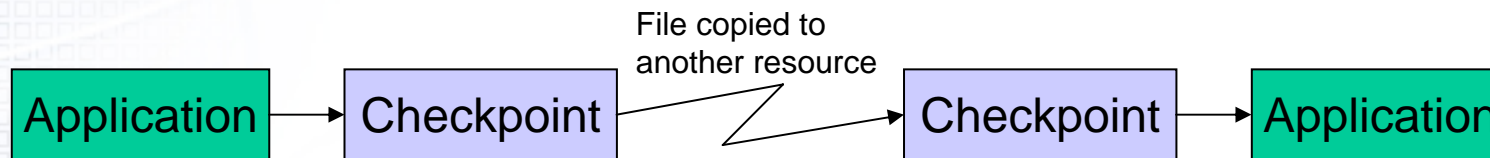


# Nimrod job dependencies

- One job cannot start until another job is finished

Examples:

- One application needs the output files from another
- An application checkpoints its data to a file to continue on another node



- One job cannot start until another job starts

Example:

- GriddLes work flow



parameter **expnr** integer range from 1 to 90 step 1;  
sequemeter **timestep** integer range from 1 to 252 step 1;

#### **task main**

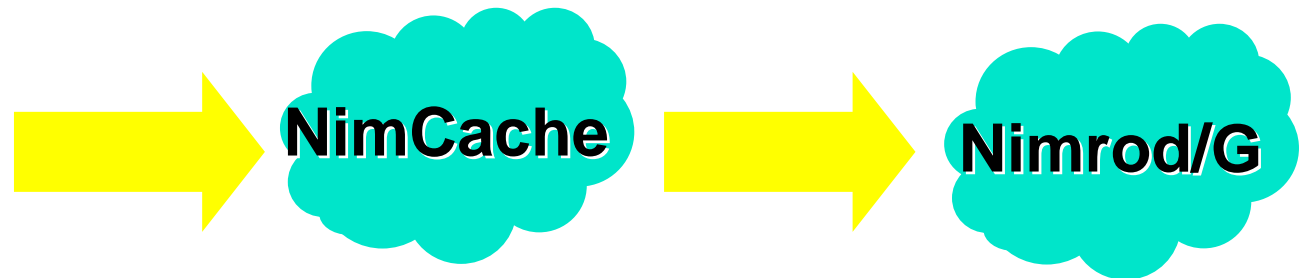
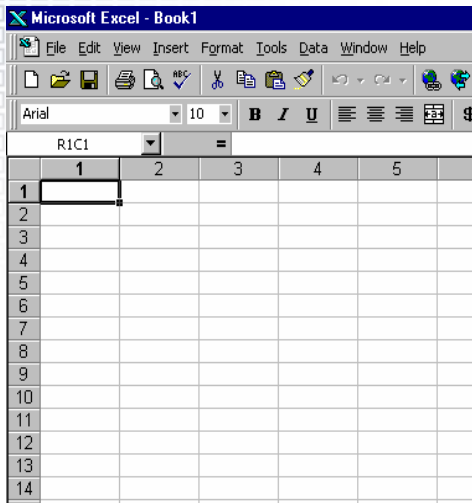
```
inputs cms.${expnr}./bin/echo [${timestep} - 1].tar.gz vf_ccam.${timestep}
outputs cms.${expnr}.${timestep}.tar.gz
sequemeters timestep
parameters expnr
node:execute tar xfz cms.${expnr}.*.tar.gz
node:execute /bin/mv vf_ccam.${timestep} cms.${expnr}/input/vf_ccam.${timestep}
node:execute cd cms.${expnr} && ./cms_prep.ksh ${expnr} ${timestep} >> ../output 2>&1
node:execute /bin/ls -l cms.${expnr}/input >> output 2>&1
node:execute /bin/ls -l cms.${expnr}/output >> output 2>&1
node:execute cd cms.${expnr}/output && . ${HOME}/bin/globpea >> .././output 2>&1
node:execute cd cms.${expnr} && ./cms_postp.ksh ${expnr} ${timestep} >> ../output 2>&1
node:execute tar cfz cms.${expnr}.${timestep}.tar.gz cms.${expnr}
```

#### **endtask**



# Active Sheets

- Users are used to using spread sheets for scenario exploration
- Existing spread sheets do not support parallel execution
- Extending MS Excel to allow execution of remote functions





# Active Sheets

Microsoft Excel - Slow Sine

File Edit View Insert Format Tools Data Window Help

Insert ActiveSheets Control Remove ActiveSheets Control Run ActiveSheets CodeBuilder Hide Current Calculation Manager Show Current Calculation Manager

B11 =default(A11,20)

**Trapezoidal Integration of External Function "SlowSine" using Nimrod over Globus** Area #####

Step size 10

X	f(x)	trap area
0	#VALUE!	#VALUE!
10	#VALUE!	#VALUE!
20	#VALUE!	#VALUE!
30	#VALUE!	#VALUE!
40	0.642788	#VALUE!
50	#VALUE!	#VALUE!
60	#VALUE!	#VALUE!
70	#VALUE!	#VALUE!
80	#VALUE!	#VALUE!
90	#VALUE!	#VALUE!
100	#VALUE!	#VALUE!
110	#VALUE!	#VALUE!
120	#VALUE!	#VALUE!
130	#VALUE!	#VALUE!
140	#VALUE!	#VALUE!
150	#VALUE!	#VALUE!
160	#VALUE!	#VALUE!
170	#VALUE!	#VALUE!
180	#VALUE!	#VALUE!
190	#VALUE!	#VALUE!
200	#VALUE!	#VALUE!
210	#VALUE!	#VALUE!
220	#VALUE!	#VALUE!
230	#VALUE!	#VALUE!
240	#VALUE!	#VALUE!
250	#VALUE!	#VALUE!

**Area of Trapezoids**

**Function Values**

**Calculation Manager - Slow Sine.xls**

Cell	Default	Time
\$B\$7:0	20	20
\$B\$8:0	20	30
\$B\$9:0	20	40
\$B\$10:0	20	50
\$B\$11:0	20	60
\$B\$12:0	20	70
\$B\$13:0	20	80
\$B\$14:0	20	90
\$B\$15:0	20	100
\$B\$16:0	20	110
\$B\$17:0	20	120
\$B\$18:0	20	130
\$B\$19:0	20	140
\$B\$20:0	20	150
\$B\$21:0	20	160
\$B\$22:0	20	170

**demo #1 - Nimrod**

File Experiment

executing

Current time	Budget
	0

Time remaining	Minimise	time

Deadline	Feasibility	feasible

Unscheduled 0/0

Resource	Usage
hammie.isi.edu	0/10
ico16.mcs.arl.gov/jobmanager-loadleveler	0/14
pitcairn.mcs.arl.gov	1/12

Ready

Start Active... Conta... C:\WI... Eudor... Micros... demo... Slow S... Calculatio... 4:14 PM



# Nimrod Java and WebServices APIs

---

- Many new application front ends and Web Portals are developed in Java
- The Nimrod Java and WebServices APIs makes it easy for developers to include Nimrod as their backend computational resource manager
  - `NimrodAPI nimrod = NimrodAPI.getNimrodAPI();`
- The Nimrod WebService API allows any programming language to use the API





# Nimrod Java API

```
// Get nimrod handle for current Unix user
```

```
NimrodAPI nimrod = NimrodAPI.getNimrodAPI();
```

```
// Add test client experiment
```

```
String[] pn = {"x", "y"};  
NimrodExperimentAPI ne =  
nimrod.addNimrodGExperiment("testClient", pn, "task main\n  
node:execute /bin/uname\nendtask", "");
```

```
// Add 100 job sweep
```

```
NimrodParameterRange[] pr = new NimrodParameterRange[2];  
pr[0] = new NimrodParameterRange();  
pr[0].parameter = "x";  
pr[0].type = "int";  
pr[0].value1 = "1";  
pr[0].value2 = "10";  
pr[0].step = "1";  
pr[1] = new NimrodParameterRange();  
pr[1].parameter = "y";  
pr[1].type = "int";  
pr[1].value1 = "1";  
pr[1].value2 = "10";  
pr[1].step = "1";  
ne.addSweep(pr);
```

```
// Add another 10 job sweep
```

```
pr[0].value1 = "12";  
pr[0].value2 = "12";  
ne.addSweep(pr);
```

```
// Register a new class for notification
```

```
nimrod.addJobListener(new  
NimrodJobListener()  
{  
    public void nimrodJobChange(NimrodJobInfo[] nji)  
    {  
        int i;  
        for (i = 0; i < nji.length; i++)  
            System.out.println("Job " + nji[i].jobname + " status " +  
nji[i].status);  
    }  
});
```

```
// Add resource to the experiment
```

```
NimrodResourceAPI nr = nimrod.getResource("fork");  
ne.addResource(nr);
```

```
// Running the experiment
```

```
ne.start();  
while (ne.getExperimentInfo().done < 110)  
{  
    System.out.println("Done: " + ne.getExperimentInfo().done);  
    try{  
        Thread.sleep(5000);  
    }catch(Exception e){  
    }  
}
```

```
// Remove the experiment
```

```
ne.deleteExperiment();
```