Nimrod projects

Colin Enticott
Slavisa Garic
Tom Peachey

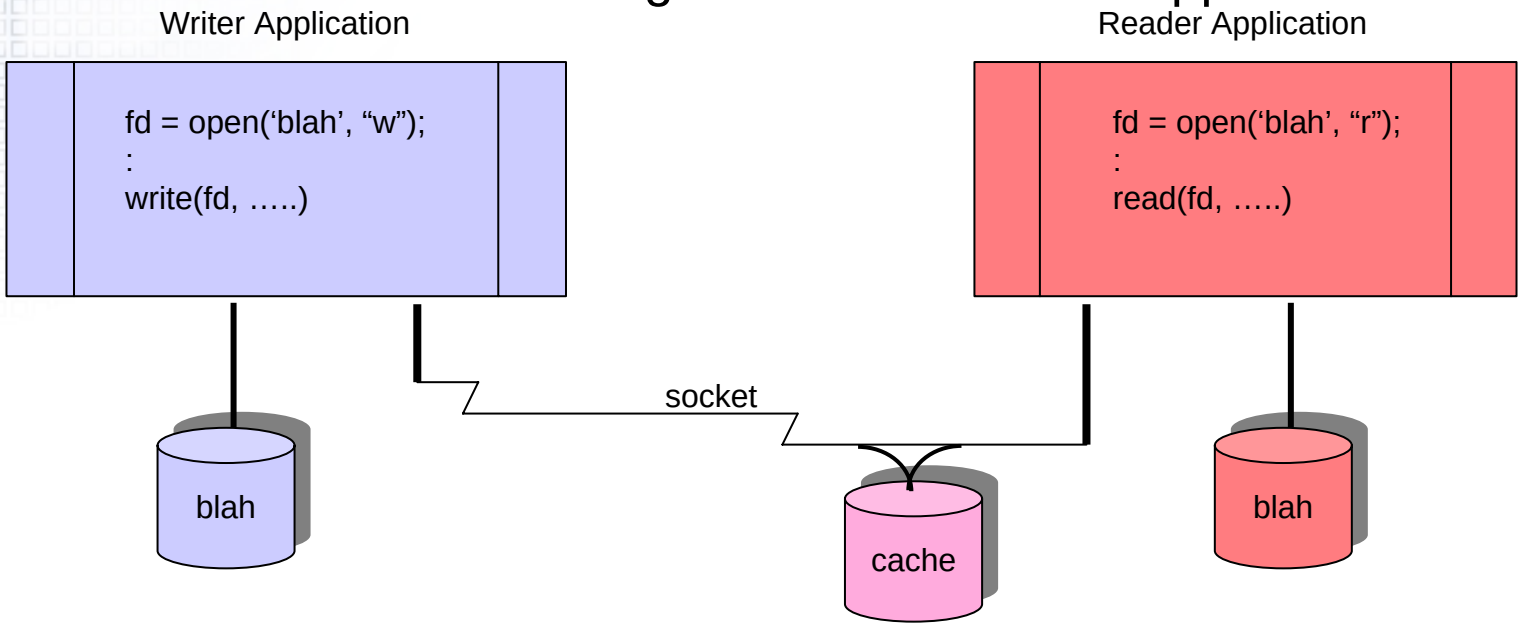
Monash University

Nimrod related projects

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

GriddLes

- http://www.csse.monash.edu.au/~davida/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



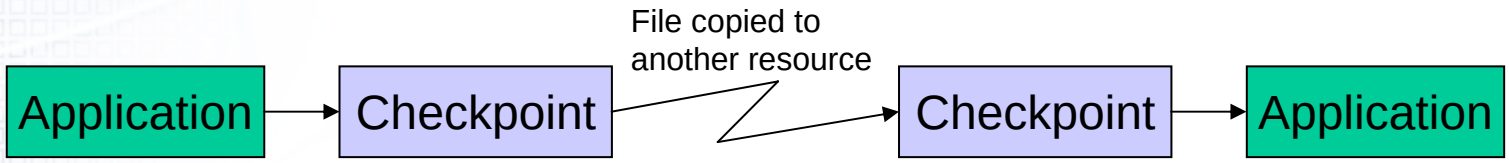
Nimrod job dependencies

- One job cannot start until another job is finished

  Examples:

  – One application needs the output files from another

  – An application checkpoints it data to a file to continue on another node



- One job cannot start until another job starts

  Example:

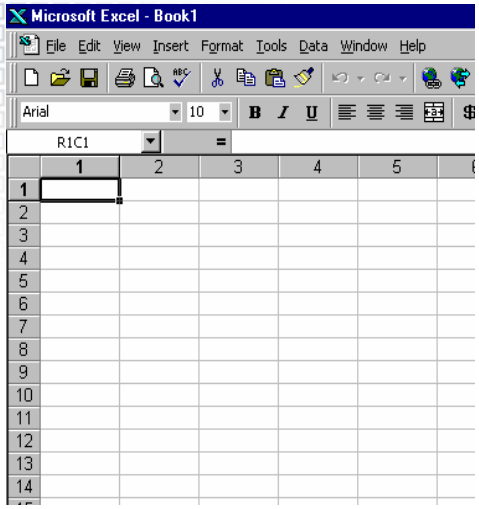
  – GriddLes work flow

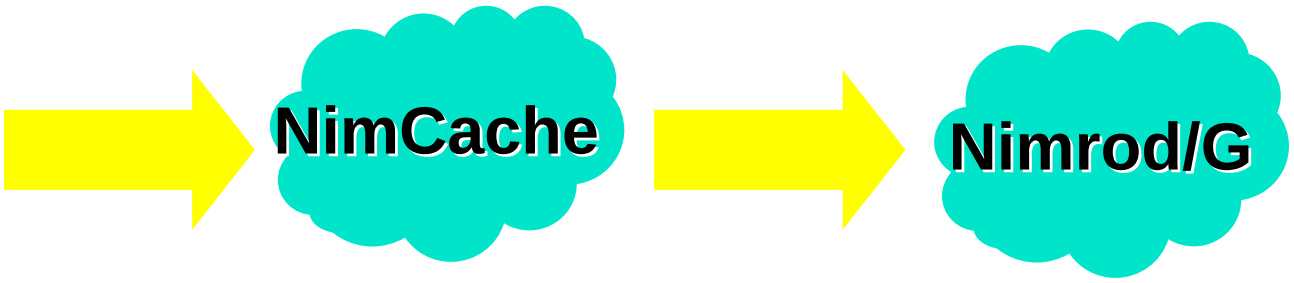
```
parameter expnr integer range from 1 to 90 step 1;
sequemeter timestep integer range from 1 to 252 step 1;

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
```

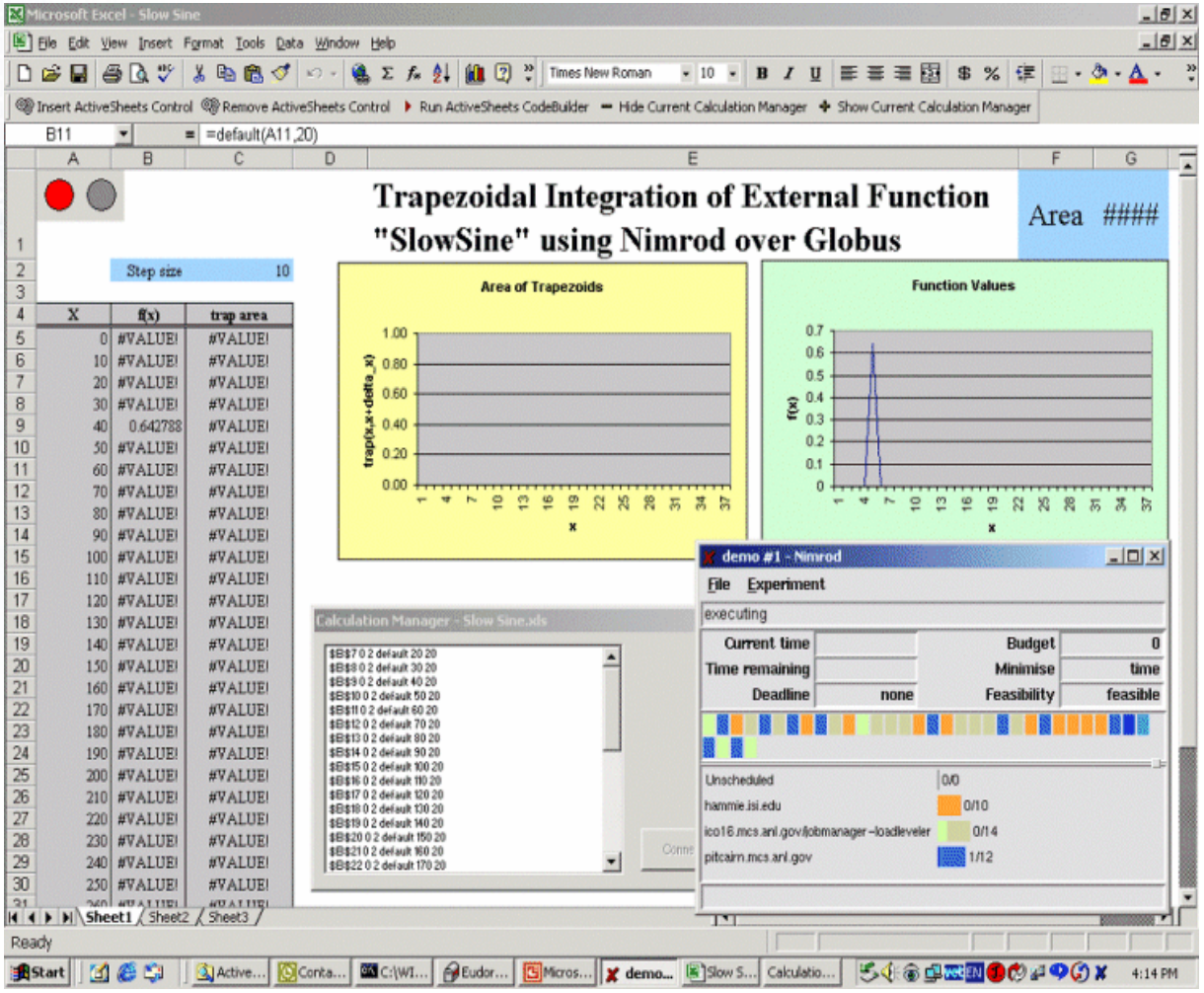
Active Sheets

- 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



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();
```