Software Engineering for the Computational Grid

David Abramson

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

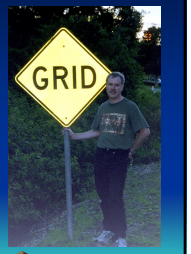
Monash University
DSTC

This talk

- What is the Grid?
- Software Engineering for the Grid
  - Software Lifecycle
- Applications Development
  - Parametric Execution
    - The Nimrod Family of tools
  - Porting legacy codes
    - GriddLeS
- Test & Debug
  - Guard
- Deployment
  - GDeploy
- Execution
  - The Nimrod Portal (Australian Nimrod Testbed – ANT)
  - Active Sheets

What is the Grid?

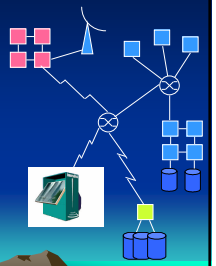
- Collection of high end resources
- Connected by high performance networks
- Scientific and other instruments
- Providing innovative distributed solutions



What does the Grid have to offer?

"Dependable, consistent, pervasive access to [high-end] resources"

- Dependable: Can provide performance and functionality guarantees
- Consistent: Uniform interfaces to a wide variety of resources
- Pervasive: Ability to "plug in" from anywhere

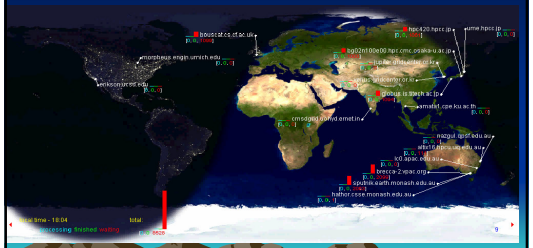


Source: www.globus.org

Fran Berman,

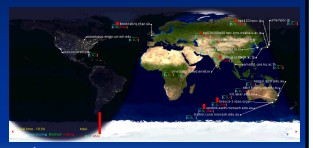
Director, SDSC, NPACI
CCGSC, Lyon, Sept 2002

What is the Grid (for me) ?



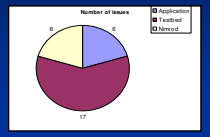
But, it aint easy!

- *Nimrod/G* to generate and distribute 50,000 instances *GAMESS*
- Ran on up to 30 super computers distributed over 10 countries.
- Spanned test beds
  - Pacific Rim Applications and Grid Middleware Assembly
  - Australian Grid Forum
  - *TeraGrid*.
- Computation ran for five days, but
  - Two research assistants worked day and night for two months before the conference to set up the infrastructure.
  - Computer scientists, not e-Scientists!



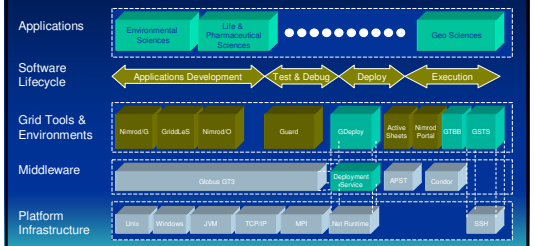
What were the issues?

- Application
  - Removing sequential assumptions
  - Building specific scripts
  - Deployment
- Testbed
  - Firewalls
  - Certificates
  - Versions of Globus
  - Installing Globus is not enough
  - "Bugs" in Globus



Software Layers and Lifecycle

Software Lifecycle



Applications Development

Applications Development on the Grid (circa 2004)

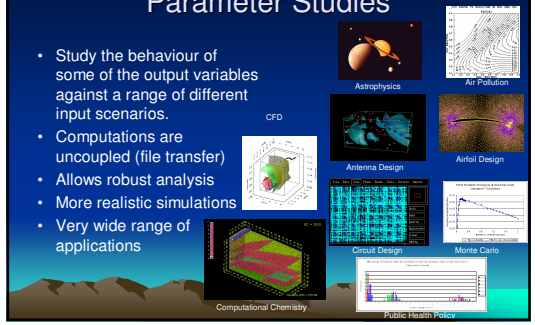
- New Applications
  - Code to middleware standards
  - Significant effort
  - Exciting new distributed application
- Legacy Applications
  - Were built before the Grid
  - They are fragile
  - File based IO
  - May be sequential
  - Leverage old codes to produce new virtual application

Reusing Legacy Components

- Parameter Sweeps (Nimrod)
  - Perform Robust Science and Engineering
  - Make distributed computing easy for this niche application
- Grid Workflows (GriddLeS)
  - Build large virtual applications across multiple organisation
  - Flexible IO model
    - File Copies
    - Pipelined computations

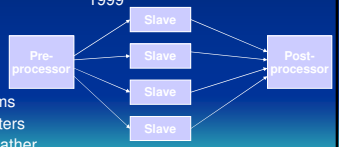
Parameter Studies

- Study the behaviour of some of the output variables against a range of different input scenarios.
- Computations are uncoupled (file transfer)
- Allows robust analysis
- More realistic simulations
- Very wide range of applications

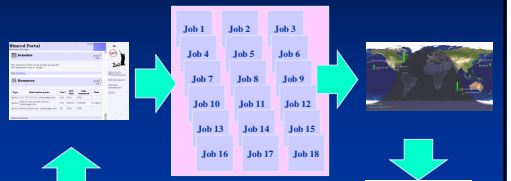


Nimrod ...

- Project History
  - Initial Cluster version 1994
  - Nimrod/G 1997
  - EnFuzion (Axceleon) 1997
  - Nimrod/O 1999
- Goals
  - Supports parametric execution
  - Execute programs
  - Varying parameters
  - Simple scatter/gather
  - Make parallel computing easy for parametric problems

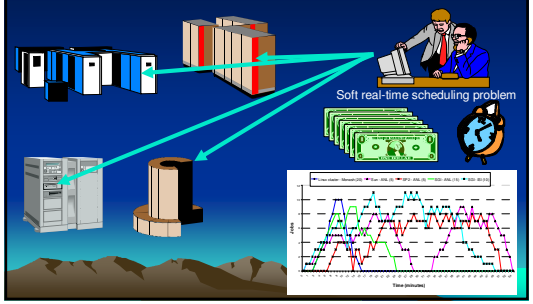


How does a user develop an application using Nimrod?

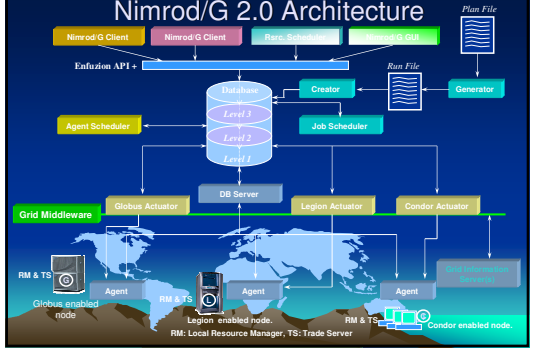


Description of Parameters PLAN FILE

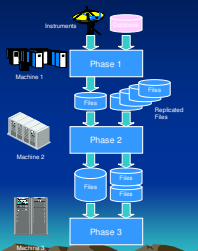
Scheduling and QOS



Nimrod/G 2.0 Architecture

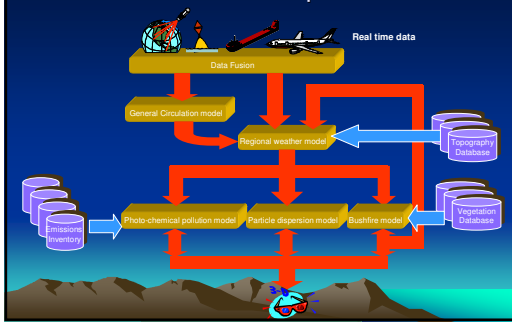


Grid Workflows

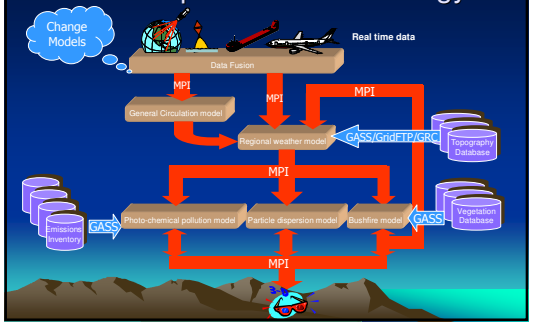


- Grid enable legacy software **without** modification
- Building virtual applications across virtual organizations
- Support pipelines in which computations are split across resources
  - Support both file copy and streamed IO
- Match computation to best resource
- Data may be distributed and replicated
- Overload file primitives to support

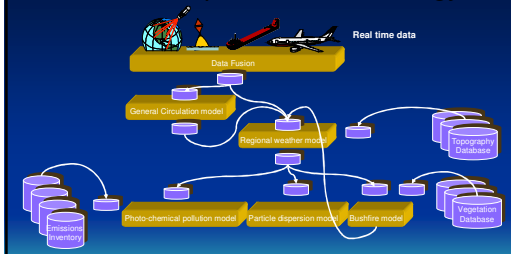
Grid Workflow - Atmospheric Sciences



One Implementation Strategy



Another implementation strategy

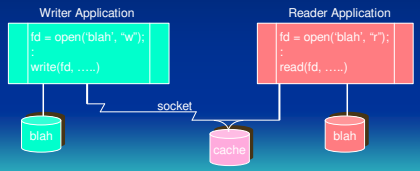


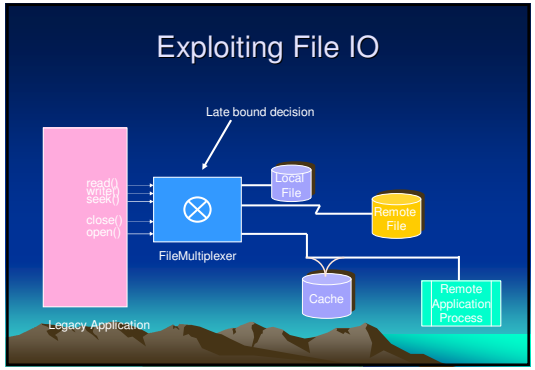
The problem is …

- Each of these implementation scenarios
  - Requires significant source level modification
  - Means architectural decisions need to be made early on and are locked in
  - Ignores existing “file” interface

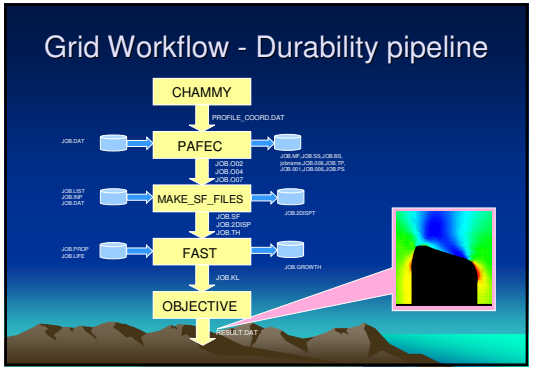
What’s wrong with file interfaces?

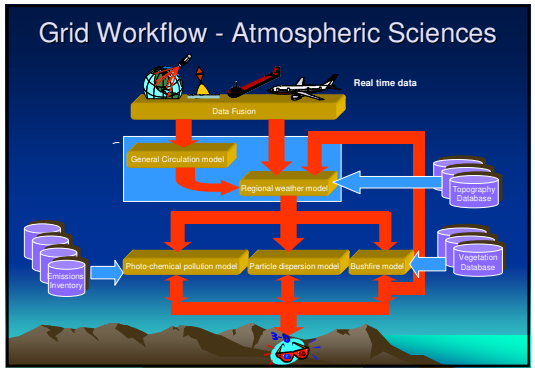
- Unix pipes are well established technique for software construction

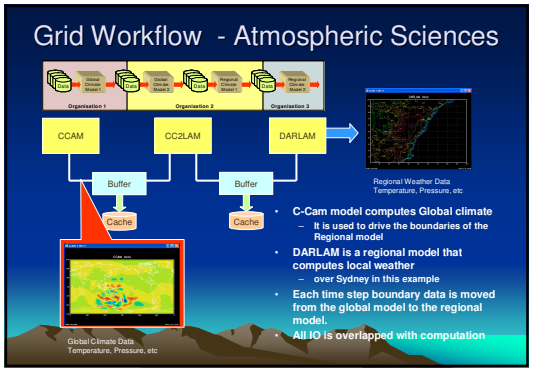


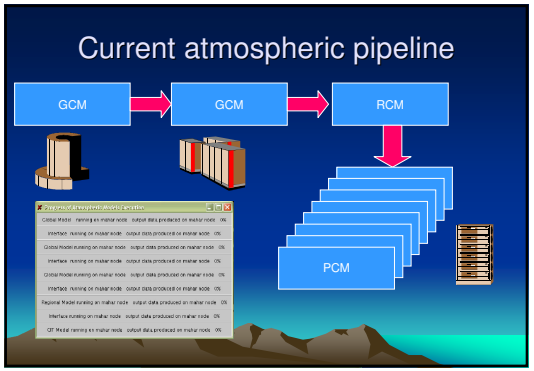

Exploiting File IO
Late bound decision
read()
write()
seek()
close()
open()
FileMultiplexer
Local File
Remote File
Cache
Remote Application Process
Legacy Application


Some examples

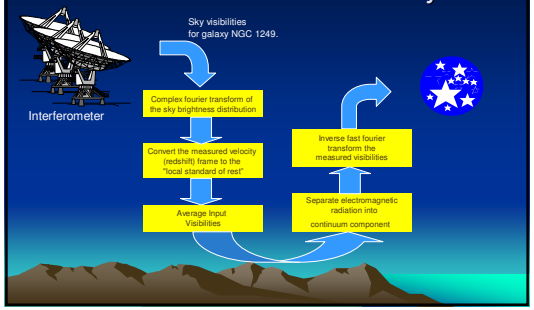

Grid Workflow - Durability pipeline
CHAMMY
PROFILE_COORD.DAT
JOB.DAT
PAFEC
JOB.C02
JOB.C04
JOB.C07
JOB.LIST
JOB.INP
JOB.DAT
MAKE_SF_FILES
JOB.SF
JOB.2DISP
JOB.TH
JOB.PROP
JOB.LIFE
FAST
JOB.KL
OBJECTIVE
RESULT.DAT


Grid Workflow - Atmospheric Sciences
Real time data
Data Fusion
General Circulation model
Regional weather model
Topography Database
Photo-chemical pollution model
Particle dispersion model
Bushfire model
Emissions Inventory
Vegetation Database

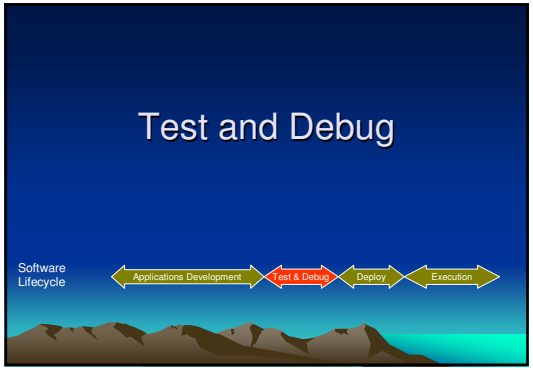

Grid Workflow - Atmospheric Sciences
Organisation 1
Organisation 2
Organisation 3
CCAM
CC2LAM
DARLAM
Buffer
Buffer
Cache
Cache
Regional Weather Data
Temperature, Pressure, etc
Global Climate Data
Temperature, Pressure, etc
C-Cam model computes Global climate
It is used to drive the boundaries of the Regional model
DARLAM is a regional model that computes local weather
over Sydney in this example
Each time step boundary data is moved from the global model to the regional model.
All IO is overlapped with computation


Current atmospheric pipeline
GCM
GCM
RCM
PCM

Grid workflow - Astronomy



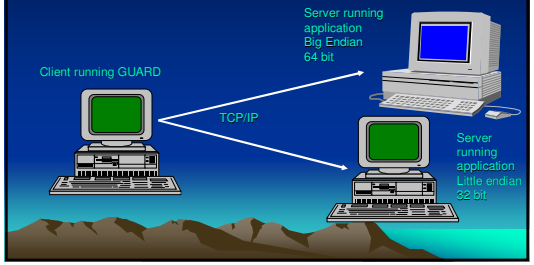
Test and Debug

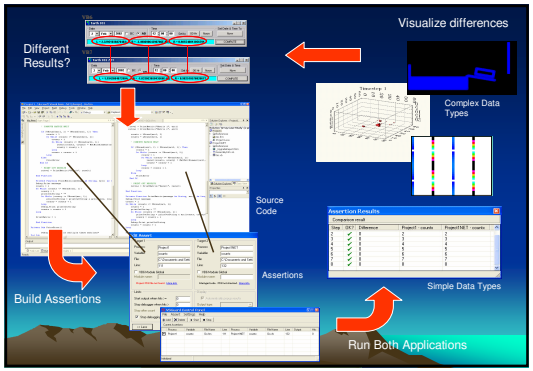


Relative debugging

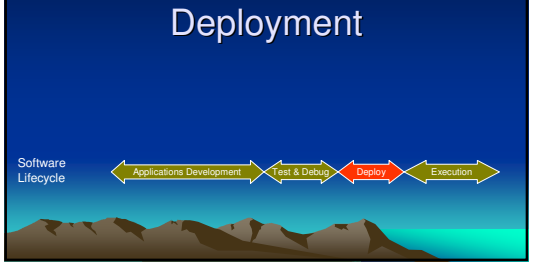
- What do you do when you move your application to another node of the Grid and it stops working?
- Subtle errors can be introduced through changes
  - By programmer
  - In the environment (DLL Hell)
- Programmer must understand application intimately to be able to locate source of errors
- Programmer can spend much time:
  - tracing program state to locate source of error
  - understanding how code changes may have resulted in errors
- Relative debugging is about automating this process.
  - Hybrid Test and Debug methodology

Relative Debugging on the Grid





Deployment

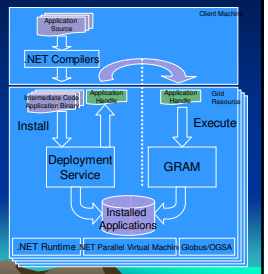


Deployment

- Globus provides functions for invocation
- Deployment is managed manually
- Deployment not easy
  - Target machines may have different instruction set architectures.
  - Different parallel architectures may be used, ranging from message passing to distributed shared memory.
  - Different operating systems or operating system versions might be used.
  - Different application libraries, or even different versions of the same library, may be installed.
  - Machine resources such as file systemsand I/O systems may differ.

Deployment Service

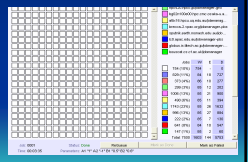
- Hide the complexity in installing software on a remote resource.
- Use local knowledge about
  - the instruction set,
  - machine structure,
  - file system,
  - I/O system, and
  - installed libraries



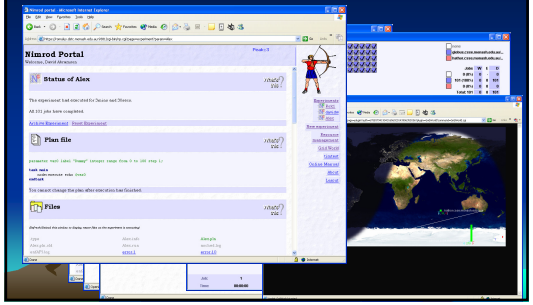
Execution

Nimrod Portal

- Provide Web interface for Nimrod/G
- Experiment creation
  - GUI and text versions of plan files
- Management of jobs
  - Stop, start & monitor
- Manage Resources
  - Certificates
  - Type of GRAM
- Schedule experiment
  - Set deadlines and budget



The Nimrod Portal

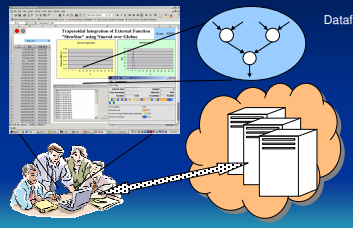


Other Interfaces

- Active Sheets
  - Spreadsheets as scripting tools
- Nimrod library
  - Web service interface
    - Callable from workflow engine
      - (Kepler, UCSD, Triana, UK eScience)
    - Callable from MatLab

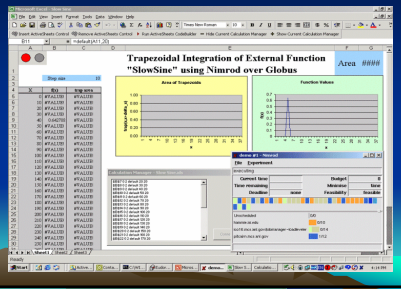
Active Spreadsheets

Script computation in spreadsheet



Dataflow graph

Active Sheets with Nimrod/G



Conclusions

- Applications Development
  - Both Nimrod and GriddLeS require no software modification
  - Integration with Grid done through wrapper
- Test and Debug
  - Relative Debugging powerful new technique
- Deployment
  - Developing a Parallel Grid Virtual Machine
- Execution
  - Web portals
  - Spreadsheets
  - Library interfaces

Acknowledgements (Monash Grid Research)

- Colleagues
  - Paul Roe, QUT
  - Jack Dongarra, UT
  - Ian Foster, ANL
  - Greg Watson, LANL
  - Raj Buyya, UoM
  - Henri Casanova, UCSD
  - Kim Baldridge, Wibke Sudholt, UCSD
- Research Fellows
  - Colin Enticott
  - Slavisa Garic
  - Jagan Kommineni
  - Tom Peachy
- Graduate Students
  - Andrew Lewis
  - Shahaan Ayyub
  - Donny Kurniawan
  - Wojtek Goscinski
  - Aaron Searle
  - Philip Chan
  - Tim Ho
- Funding & Support
  - CRC for Distributed Systems Technology
  - Australian Research Council
  - GrangeNet
  - APAC
  - Microsoft
  - Sun Microsystems
  - IBM
  - Hewlett Packard
  - Axceleon

Further Information

- Nimrod Family
  - www.csse.monash.edu.au/~davida/nimrod
- GriddLeS
  - www.csse.monash.edu.au/~davida/griddles
- Guard
  - www.csse.monash.edu.au/~davida/guard
- DSTC
  - www.dstc.edu.au
- Globus
  - www.globus.org
- Axceleon
  - www.axceleon.com