



eMinerals community grid

user-based grids for collaborative science

Martin Dove
University of Cambridge



Individual collaborators

Cambridge: Peter Murray-Rust, Emilio Artacho, Richard Bruin, Ian Frame, Gen-Tao Chiang, Stuart Ballard, Andrew Walker, Kat Austen, Toby White, Andrew Walkingshaw

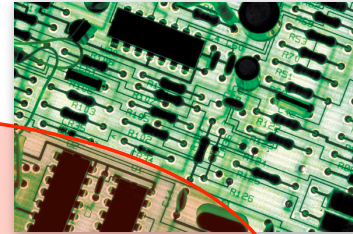
CCLRC: Rik Tyer, Kerstin Kleese van Dam, Tom Mortimer-Jones, Ilian Todorov

Bath: Steve Parker, Arnaud Marmier, Corinne Arrouvel

Reading: Vassil Alexandrov, Gareth Lewis, Ismael Bhana

Grid computing

Computing
grids



Data
grids



Collaborative
grids





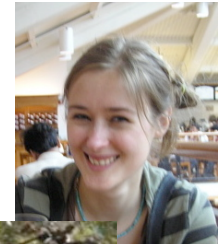
A scientist's – anarchic – view of science

There are many valid perspectives, from the *user* to the *provider* of resources

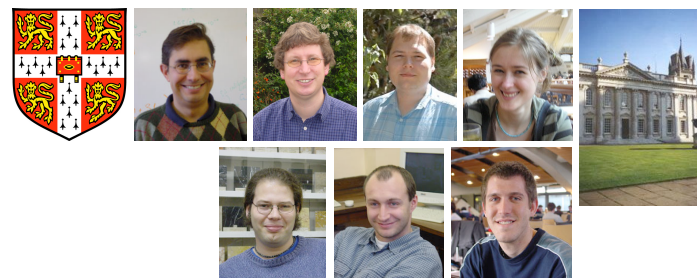
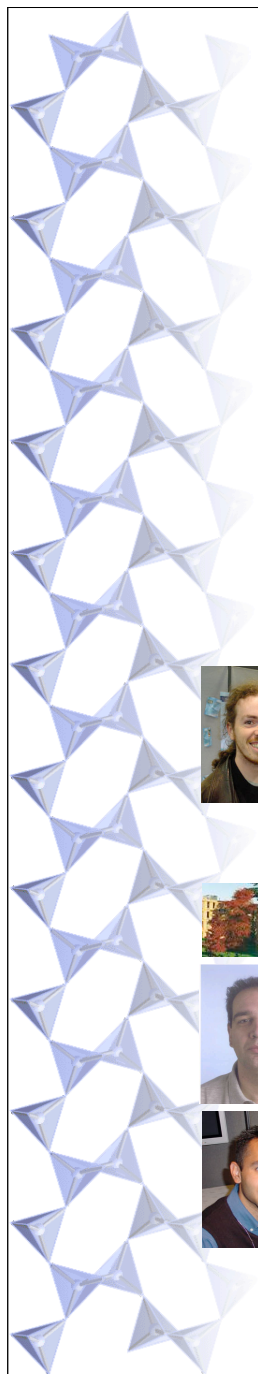
- The eMinerals approach is to focus on scientists: *their* work, *their* data and *their* collaborations
- Tool development is tensioned against what the scientists use
- Virtual organisations may resolve a number of technical issues pragmatically
- Scientists get their hands dirty
- Our grids are self-managed: community grids

User profile

- Our users only want portals/GUIs for specific tools, not for the working environment
- Users do not want their applications pre-wrapped as services: they want to have complete control over their applications, e.g. to add capability
- Users do not want a provider/consumer model: that does not provide the freedom they need
- Hence a bottom-up approach



The eMinerals project team



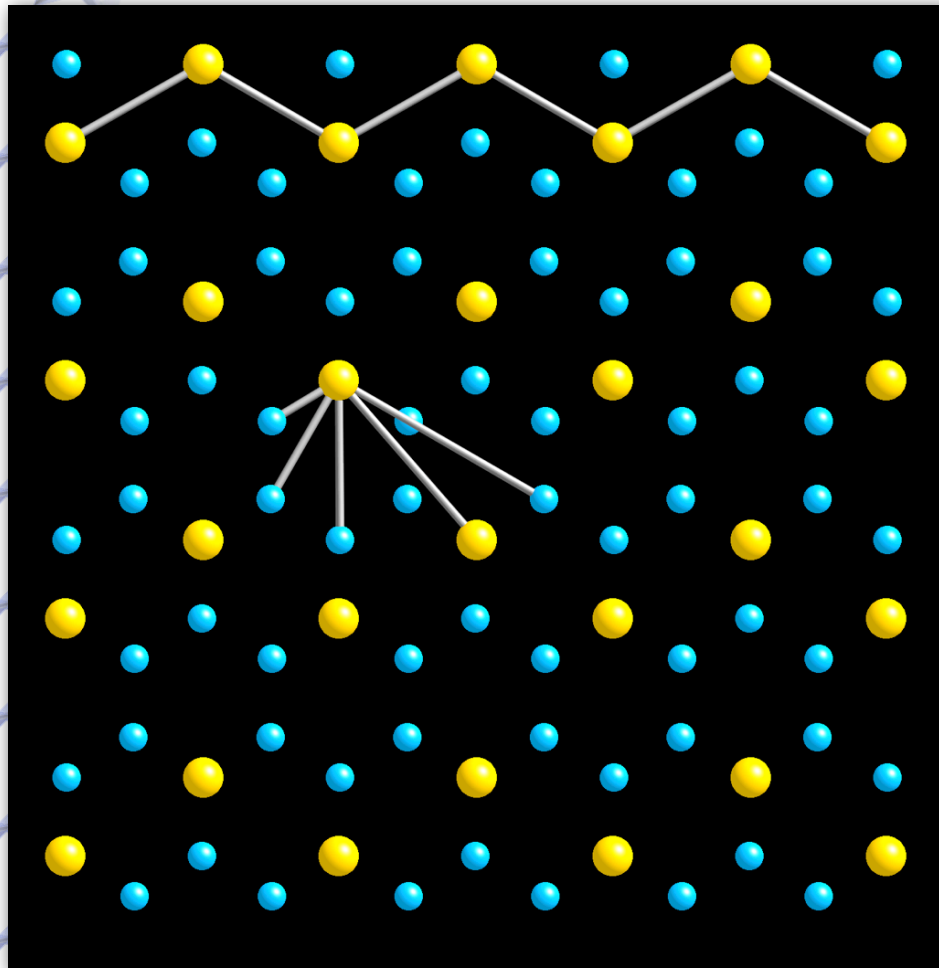
~~CCLRC~~ connection



The collaboration with the eScience centre within STFC may lead to our tools being deployed within Diamond and ISIS



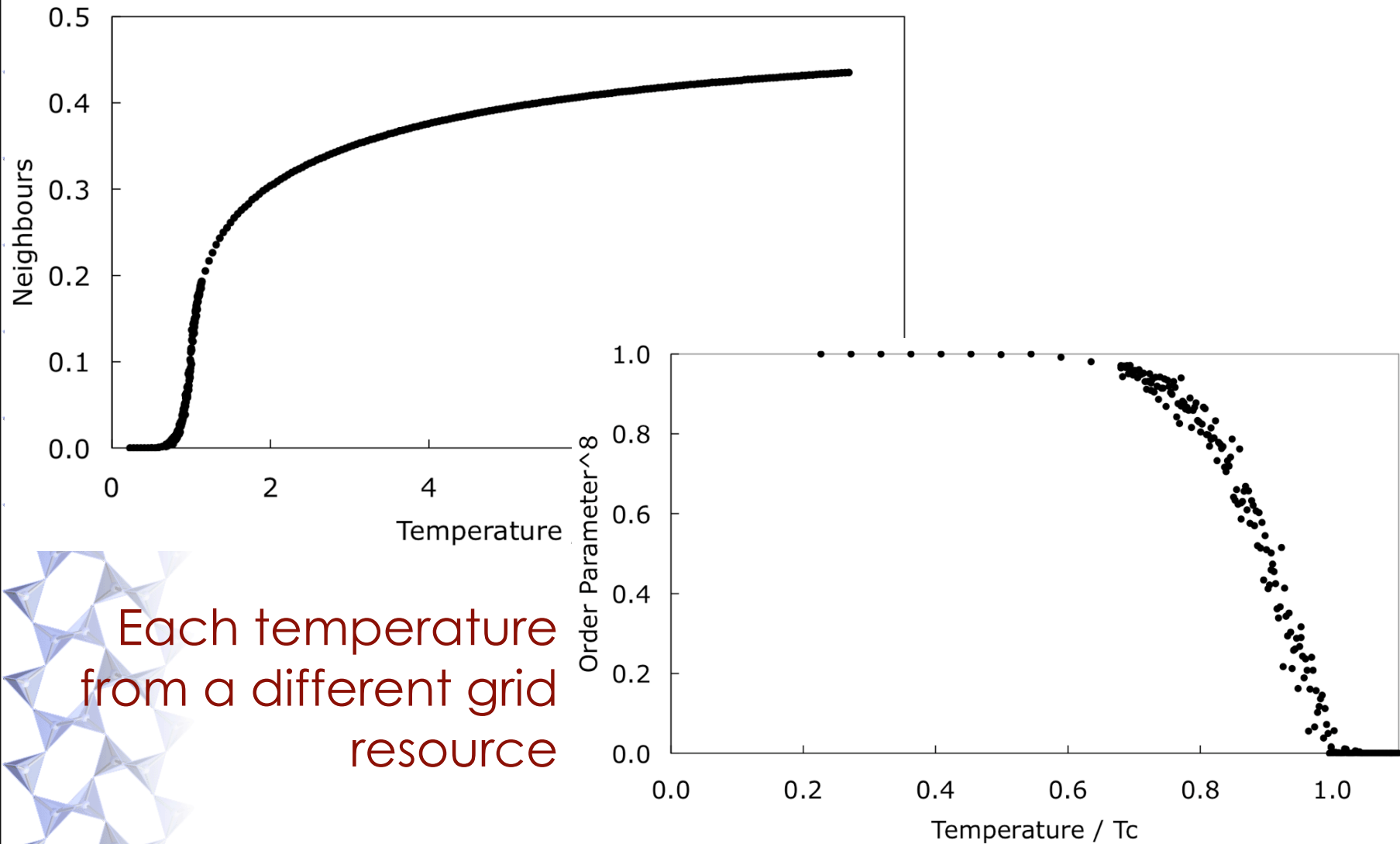
Monte Carlo simulations of cation ordering in layer silicates



Simulations of cation ordering in layer silicates based on parameterised Hamiltonian

Example of parametric study which suits grid computing environments well

Examples of results



eScience: “Science beyond the lab book”

- Management of too many computing tasks
- Management of the resultant data deluge
- Sharing the information content with collaborators




eScience can help the human scientist cope, including maintaining accuracy and accountability

Compute grids: some of the components



- High-throughput and high-performance clusters



- Pools of individual machines linked together by “Condor”



- Authentication & authorisation, and job submission, handled by “Globus”

Our community grid

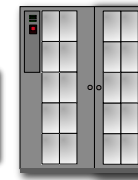
Access to external facilities and grids

Campus grids



Condor
JobMgr
Globus

Parallel (HPC) clusters



Cluster
JobMgr
Globus

Internet

Compute clusters



Cluster
JobMgr
Globus

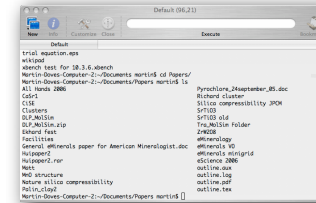
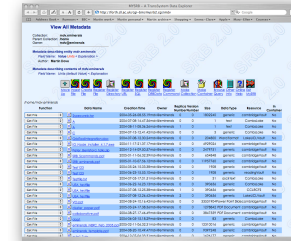
Desktop pools



Condor
JobMgr
Globus



Application
server

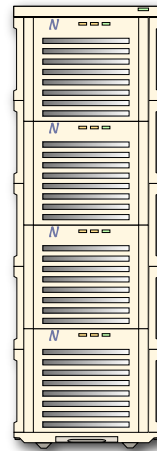


Researcher



Data grid: the San Diego Storage Resource Broker

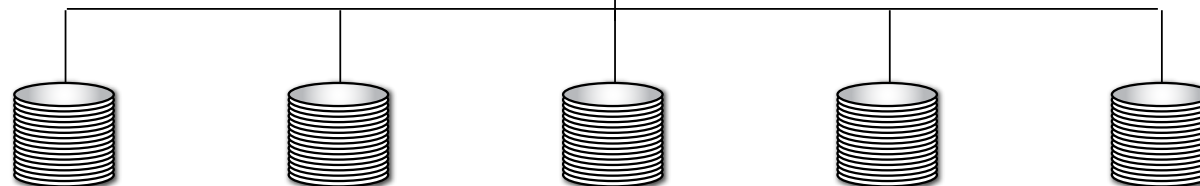
Metadata
catalogue



Internet



Distributed file
management

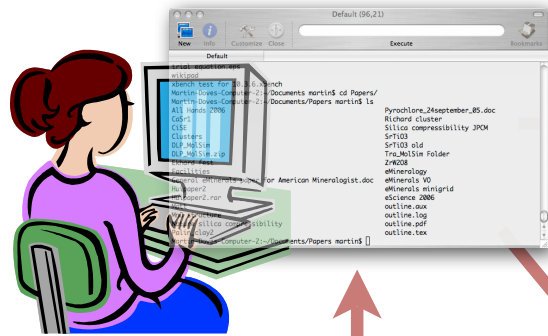


Distributed data vaults



What the SRB has given us

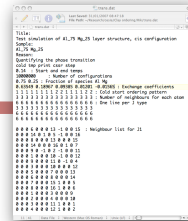
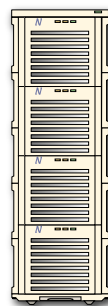
- Our scientists now expect to be able to share their data with collaborators ...
- ... and they now expect this to be easy (ie not via a multi-stage process)
- Our scientists now routinely produce complete archives of files associated with a study easily and automatically
- We now expect a single place to deposit data, and for this process to be easy and automatic



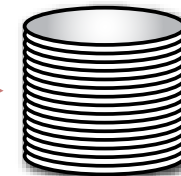
Researcher

7. Researcher interacts with the metadata database to extract core output values

Application server

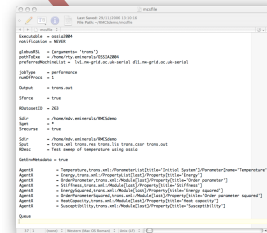


1. Upload data files and application to data vault

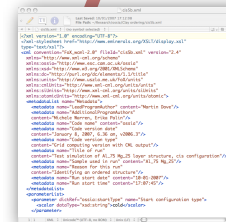


Data vault

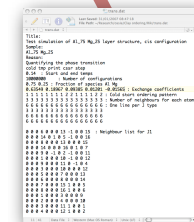
3. Data files and application are transferred to the grid resource



2. Submit job to minigrid via RMCS



6. Output files are transferred to the data vault

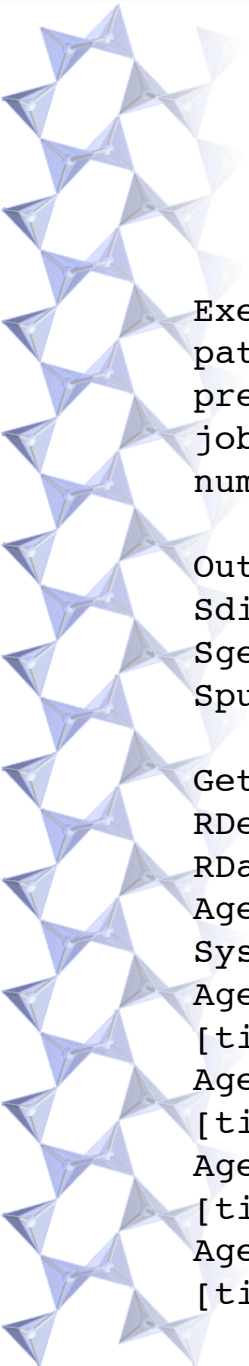


4. Job runs on grid compute resources



5. Metadata is sent to the application server

Our user interface



```
Executable          = ossia2004
pathToExe           = /home/bob.eminerals/OSSIA2004
preferredMachineList = lvl.nw-grid.ac.uk-serial dll.nw-grid.ac.uk-serial
jobType             = performance
numOfProcs          = 1

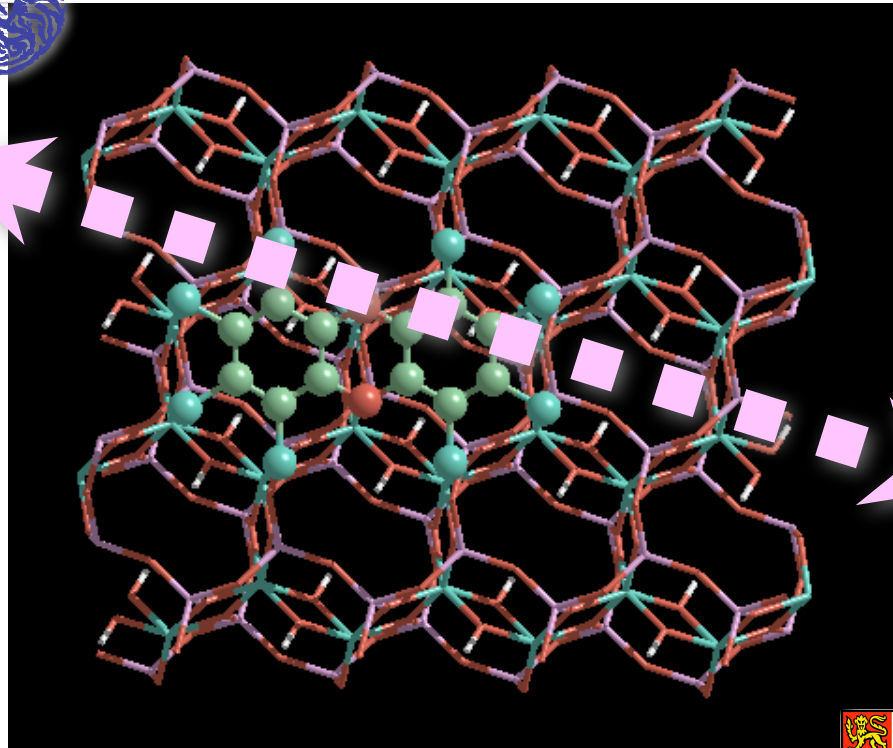
Output              = trans.out
Sdir                 = /home/bob.eminerals/RMCSdemo
Sget                 = *
Sput                 = *

GetEnvMetadata      = true
RDesc                = Test sweep of temperature using ossia
RDatasetID          = 263
AgentX               = Temperature,trans.xml:/ParameterList[title='Initial
System']/Parameter[name='Temperature']
AgentX               = Energy,trans.xml:/PropertyList[last]/Property
[title='Energy']
AgentX               = OrderParameter,trans.xml:/Module[last]/Property
[title='Order parameter']
AgentX               = HeatCapacity,trans.xml:/Module[last]/Property
[title='Heat capacity']
AgentX               = Susceptibility,trans.xml:/Module[last]/Property
[title='Susceptibility']
```

Scientific collaboration



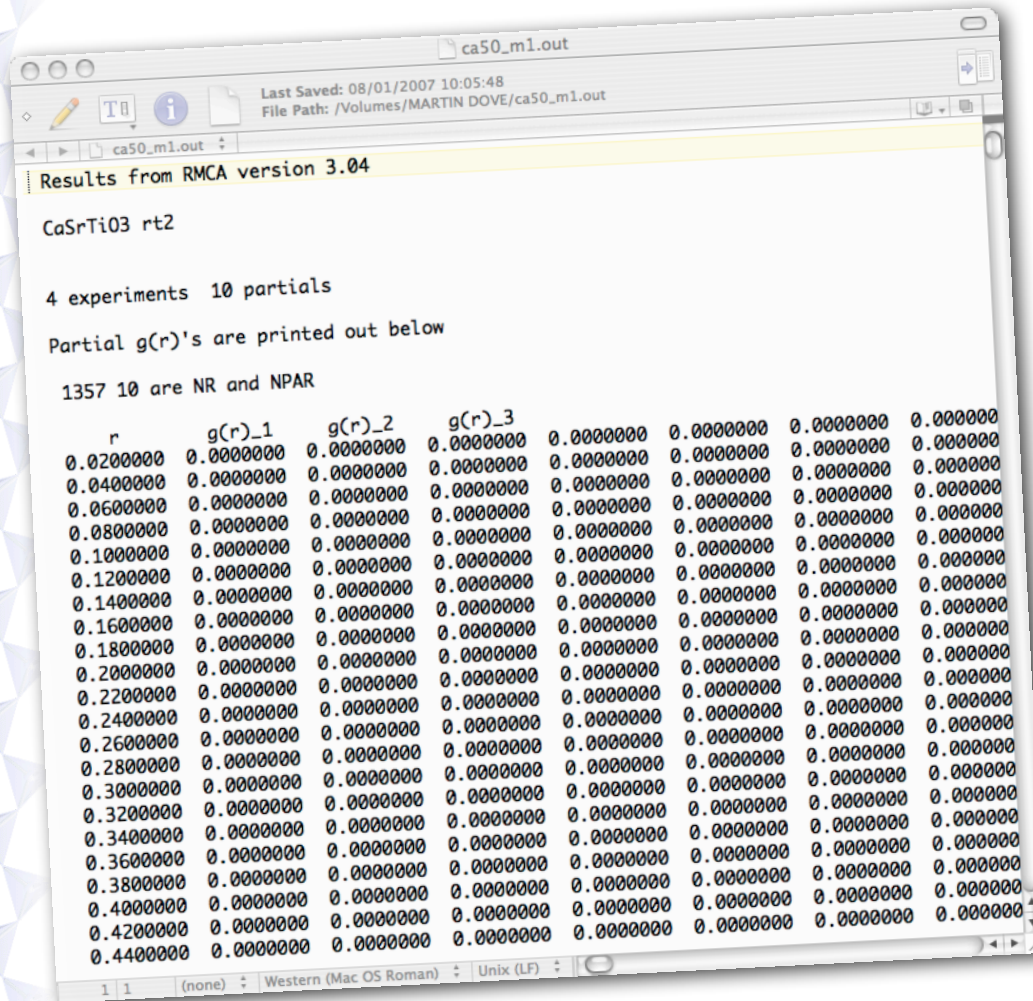
Classical
molecular
dynamics
methods



Quantum
mechanical
methods



Data and information



ca50_m1.out

Last Saved: 08/01/2007 10:05:48
File Path: /Volumes/MARTIN DOVE/ca50_m1.out

Results from RMCA version 3.04

CaSrTiO3 rt2

4 experiments 10 partials

Partial $g(r)$'s are printed out below

1357 10 are NR and NPAR

r	$g(r)_1$	$g(r)_2$	$g(r)_3$				
0.0200000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.0400000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.0600000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.0800000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.1000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.1200000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.1400000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.1600000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.1800000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.2000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.2200000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.2400000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.2600000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.2800000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.3000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.3200000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.3400000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.3600000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.3800000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.4000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.4200000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.4400000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

1 1 (none) Western (Mac OS Roman) Unix (LF)





Data and information sharing: XML data representation

```
<?xml version="1.0" encoding="UTF-8"?>
<cml convention="FoX_wcml-2.0" fileId="cis1.cml"
  version="2.4" xmlns="http://www.xml-cml.org/schema">
```

Chemical Markup Language

```
<metadataList name="Metadata">
  <metadata name="Code name" content="ossia"/>
  <metadata name="Code version date" content="January 8, 2007, v2007.3"/>
  ...
</metadataList>
```

Capturing audit metadata

```
<module title="Initial System" dictRef="emin:initialModule">
  <parameterList>
    <parameter dictRef="ossia:temperature" name="Temperature">
      <scalar dataType="xsd:double" units="cmlUnits:eV">1.000000000000e-1</scalar>
    </parameter>
    <parameter dictRef="ossia:NumberOfSteps" name="Number of steps">
      <scalar dataType="xsd:integer" units="units:countable">10000000</scalar>
    </parameter>
    ...
  </parameterList>
</module>
...
```

Capturing initial parameters

```
<module title="Finalization" dictRef="emin:finalModule">
  <propertyList>
    <property dictRef="ossia:Energy" title="Energy">
      <scalar dataType="xsd:double" units="cmlUnits:eV">2.052516362912e-1</scalar>
    </property>
    ...
  </propertyList>
</module>
</cml>
```

Capturing computed properties



XML and Fortran

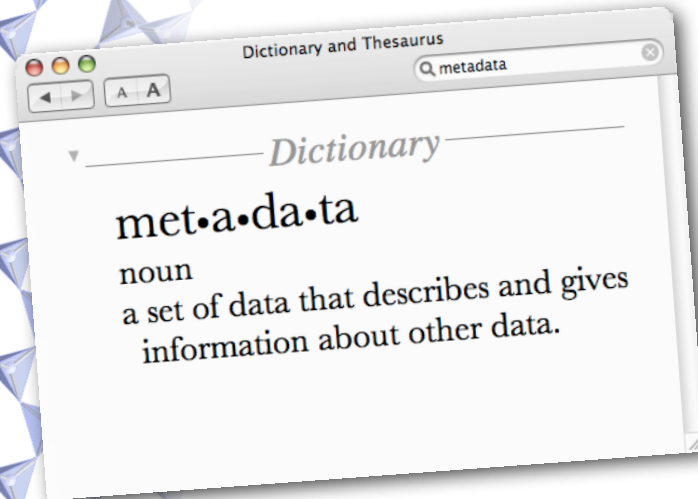
- Most of our simulation codes are written in Fortran, which has little support for XML
- Thus we have written a set of XML libraries for Fortran – called FoX – which make writing XML easy
- We have XML-ised a number of simulation codes, including SIESTA, CASTEP, DL_POLY and GULP
- We have also developed an XML-aware interface to the SRB called TobysSRB



What XML gives us

- Simulation code output that is self-describing (no more mere lists of numbers!)
- XML files can be transformed to give user-centric and information-centric representations of data, including plotted data
- XML files can have key information extracted easily, essential for large combinatorial studies
- XML enables automatic capture of metadata, and metadata is essential for managing data

XML → metadata



- Our job submission tools automatically harvest metadata from our output XML files
- We have developed a new set of tools to access the metadata database (“RCommands”)
- We use metadata for locating data and datasets created by our colleagues
- We also use metadata for extracting core information from data – useful for analysing combinatorial studies



RCommands and metadata

Metadata are associated with a hierarchy of studies, datasets and data objects, both as descriptions and as name/value pairs.

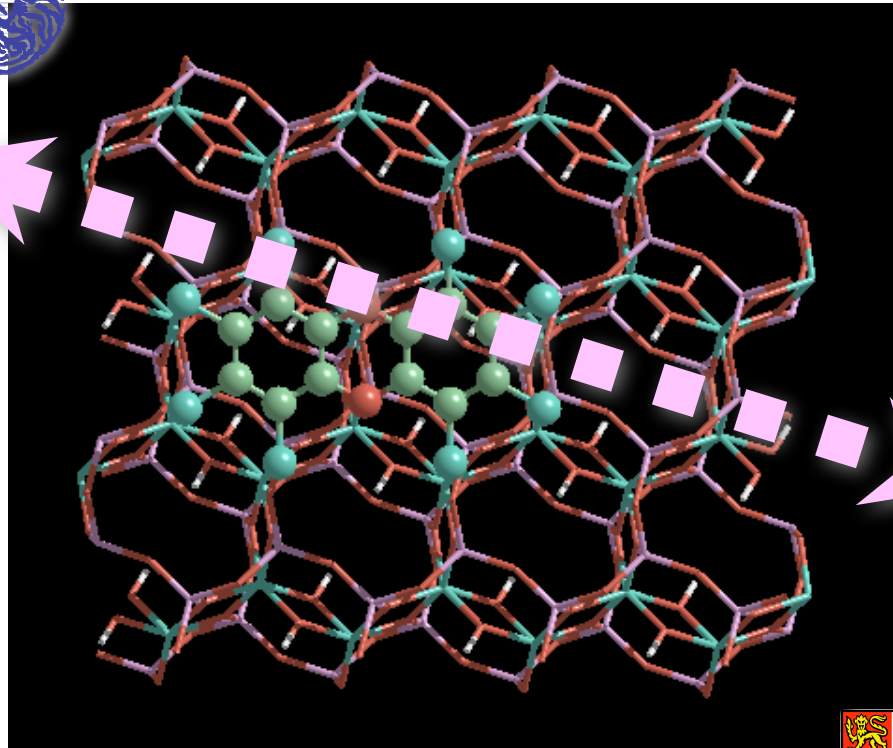
Examples of commands:

- Rls: list metadata items
- Rget: get metadata
- Rannotate: add metadata
- Rgem: extract metadata from all data objects within a dataset

Scientific collaboration



Classical
molecular
dynamics
methods



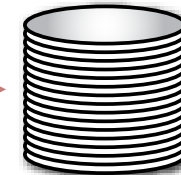
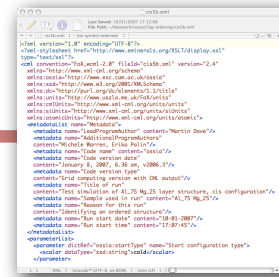
Quantum
mechanical
methods



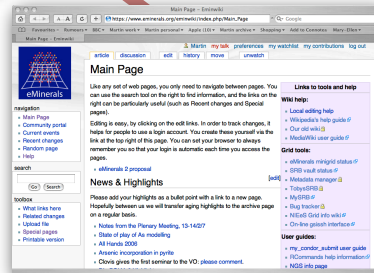


Researcher A

Upload XML data files to data vault for sharing with collaborator



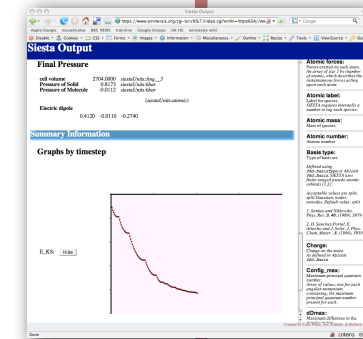
Data vault



Project wiki



Instant messaging



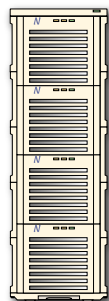
Annotate data with metadata

Access Grid with JMAST

email

Using Rger simulation outputs

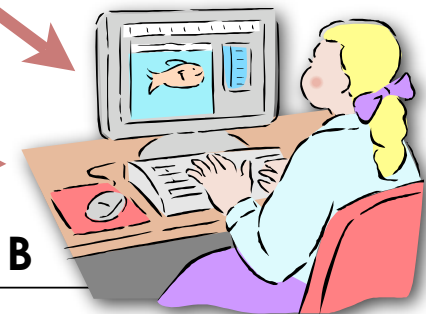
View information content of data files using ccViz



Application server

Locate data from metadata

Researcher B





Social networking sites, Web 2.0

These have the potential to revolutionise how scientists work

- MySpace, facebook etc ..
- <http://nature.network.com/>
- <http://www.scispace.net/>



Summary

- Grid computing is radically changing how we do simulation science
- I have discussed a community-grid approach based on integration of compute and data grids, with extensibility, and a focus on aiding collaboration
- I am happy to demonstrate some of this stuff within the UK eScience village