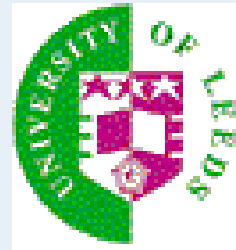




A Web Services Infrastructure for General Computing

Aaron Turner,
University of York / White Rose Grid





The White Rose Grid

- UK based academic Grid:
 - Leeds, Sheffield, York.
 - Circa 500 CPUs.
- E-Science Centre of Excellence.
- Close links to Sun.
- Hosts a National Grid node.
- Commercial spin off management.



Introduction

- Not a finished solution.
- Intended for discussion.
- Intended to feed into GGF, etc.



What is General Computing?

- The ability to submit any job to a machine
 - Not just ones using executables already on remote machine.
- Practically, limit to compatible binaries.
 - Could use virtual machines to allow multiple OS compatibility in theory.
- Often home grown codes.



Why a Web Services Interface?

- Common for 'canned' executables:
 - Executables which are fixed, but with some or all data on which it operates supplied by the user.
- De facto standard.
- Fits into work on work flow management



Interface

- Advertises capability to run general computing jobs.
- From underlying DRM:
 - Advertises memory, disk, available, etc.
 - Advertises scheduling information.
 - Architectures supported.



The Dependency Problem

- A binary may depend on:
 - Libraries.
 - Executables.
 - Interpreters.
 - Hardware.
- Possible to publish machine capabilities and to provide a matching service
 - Limits machines on which a program may run.
 - May exclude otherwise suitable machines.
 - May be no match.



Relation to JSDL 0.3

- JSDL 0.3 has weak support for describing dependencies.
- Systems to allow satisfaction of them lies outside JSDL.



Potential Solutions to the dependency Problem

- Create a description of the dependencies
 - Fold this into JSDL.
- Create mechanisms to allow a remote machine to satisfy these dependencies:
 - Use existing tools at the low level where available:
 - Sun package management.
 - Apt-get.
 - Red Carpet.
 - others.



Dependency Languages

- Various tools for describing dependencies exist:
 - System/OS specific.
 - Need a generalised one.
- Needs to be able to capture a lot of information.
- Tools need to be developed to automate describing systems to capture the dependencies:
 - Can't rely on users.
- Need to be able to allow users to pick up dependencies on other things they have written.
- Needs to be licensing aware, ideally.



System Admin Concerns

- Concerns about machine stability if libraries and executables can be loaded.
 - Machine stability.
- Bandwidth required to load libraries and executables to satisfy the dependencies.
- General concerns about allowing general computing.



Alleviating Concerns

- Install executables and binaries local to the user.
- Use virtual servers/machines.



Local User Installs

- Fairly simple.
- Need to enable writing of shell scripts, job scripts, etc. (depending on DRM) to correctly set environment variables to pick up correct dependency satisfactions.
- Could be wasteful of disk space if not handled intelligently:
 - Satisfactions should be cached to a central location for use by multiple users, or multiple times.



Virtual Servers

- An instance of the operating system is created as a sandbox:
 - Linux virtual servers.
 - Solaris 10 Zones.
 - Even virtual machines, e.g. VMWare.
- More overheads than local user installs.
- More secure due to sand boxing.



The Need for Caching

- Reduce disk space usage by linking to a central cache.
- Increases it by caching possible out of date executables and libraries.
- Allows reduction in bandwidth.
- Need strategies for removing files too.
- Licensing issues.



System Admin Concerns 2

- Main concerns are:
 - Buggy code.
 - Malicious code.
 - Buggy libraries, etc.
- Virtual servers offer the best solution to these, at the cost of overheads.
- Possibly offer multiple services based on trust:performance model?



An Overall Solution

- A web service advertises the capability to run user provided executables:
 - Publish details of architectures supported.
 - Latency for start-ups, etc.
- Job description provides descriptions of dependencies and how to satisfy them.
- Remote system goes about the job of satisfying the dependencies.
- DRM at remote site finally schedules the job to run.



Conclusion

- Make a consistent interface for fixed executables and the capability to run general computing jobs.
- Address the problem of system dependencies
 - Allows programs to run in more location.
 - Look at best strategies for this.
- Address other system administrator concerns.
- Feed back into GGF.
- Implement it.