

ReQUeST

Resource Quantification in e-Science Technologies

www.lfcs.ed.ac.uk/mrg

Overview

Computational grids integrate the services and resources of distributed, heterogeneous virtual organisations. Managing the allocation of computational resources to specific application tasks represents one of the core challenges for providing reliable, flexible and scalable grid applications.

In order to facilitate the development of such allocation managers, work is under way to equip application programs with certificates specifying the type and quantity of resources which are allocated throughout the program execution.

These certificates take the form of mathematical proofs which are derived with little or no programmer intervention by program analysis and are communicated between grid components as an integral part of the application code.

Resource allocation in the Grid

Most Grid services are offered via the Open Grid Services Architecture (OGSA), based on the Globus toolkit and web services. Globus includes an architecture for reservation and allocation (GARA) and a resource allocation manager (GRAM).

Resource requirements (computing time, memory consumption, . . .) are described in a specification language for resources (RSL), but are often based on guesses or implicit assumptions.

Current approach based on Java

Code is executed in a controlled environment (sandbox, protection domain)

which dynamically enforces access restrictions, supported by a security manager inside the class-file loader. When a violation of a resource bound is discovered during execution, all that can be done is to terminate the whole program.

However, by this time valuable resources have already been consumed and important data may have been overwritten.

Certifying resource requirements

As an alternative we aim to equip programs with proofs certifying their resource requirements. In comparison with abstract complexity theory, which deals with asymptotic resource bounds, we envisage a more fine-grained and exact quantification.

Certificates are derived from high-level type systems by resource aware compilation and are expressed in low level program logics for (virtual) machine code in which memory usage, execution time and other costs can be directly expressed.

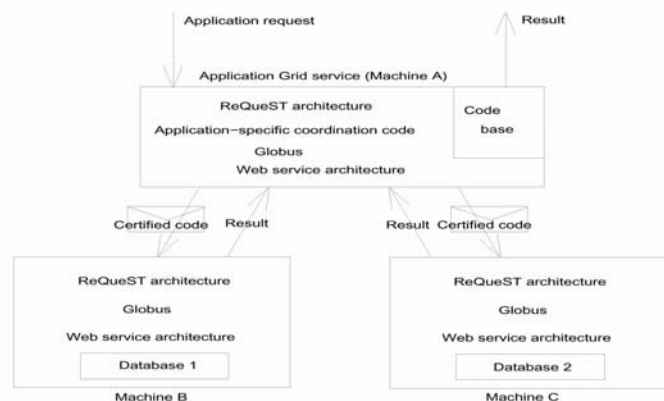


Figure 1: Grid service with mobile code

Such certification is beneficial for both service providers and application developers:

- service providers can optimise the scheduling of their service provision on the basis of accurate *a priori* information about processing costs. Applications with high resource requirements may be delayed or entirely refused permission to execute. Certified code need not be dynamically monitored.

- application developers can investigate and optimise the resource usage of their code based on statically acquired information, and may choose the most appropriate (cheapest) service suitable for their task. Certified code is guaranteed not to be aborted by the service provider during runtime due to the exhaustion of resources.

As the certificates concern inherent properties of application code, the approach complements issues of authorship certification or data encryption.

Code mobility

Certified resource usage quantification is compatible with the proof-carrying code (PCC) paradigm which allows grid computations to proceed by transmitting application programs to where the data resides, rather than transmitting data to where the program resides (see Figure 1, previous page).

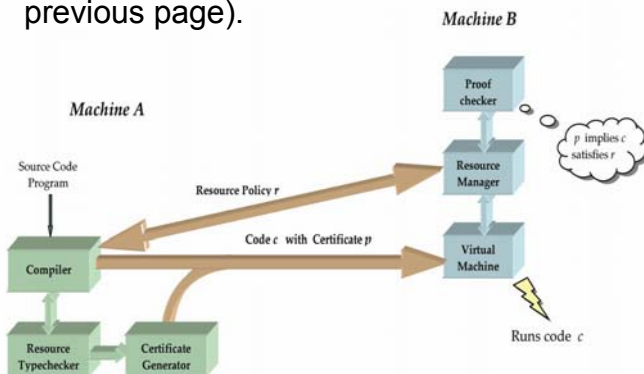


Figure 2: Code transfer and certification

Each code transmission involves shipment of certified code from a code producer (in our case Machine A) to a code consumer (Machines B and C).

Before servicing a request, each code consumer validates that the certificate matches the code's behaviour and that it is compatible with his local resource policy (see Figure 2, below left).

Current work

We are currently developing the infrastructure needed to implement the above scenario*.

We have identified a variation of Java byte-code which is suitable for the development of resource logics and are currently extending an existing compiler to transform high-level type annotations into low-level certificates.

In the future we aim to extend our framework to other high-level languages and Grid specific applications, to integrate it with the web services architecture and the Globus toolkit, and to incorporate other security and program analysis mechanisms.

Project members

D. Aspinall, L. Beringer, S. Gilmore, K. MacKenzie, A. Momigliano, D. Sannella, I. Stark

*This research is part of the MRG project (IST-2001-33149), which is funded by the EC under the FET proactive initiative on Global Computing.

