A RESOURCE-ORIENTED DATA MANAGEMENT ARCHITECTURE FOR NANOCMOS ELECTRONICS

B. Harbulot[1], D. Berry[3], C. Davenhall[3], M. Jones[1], C. Millar[2], G. Roy[4], R. Sinnott[2], G. Stewart[2], A. Asenov[4]
[1]*e-Science North West, University of Manchester*
[2]*National e-Science Centre, University of Glasgow*
[3]*National e-Science Centre, University of Edinburgh*
[4]*Department of Electronics and Electrical Engineering, University of Glasgow*
bruno.harbulot@manchester.ac.uk

Abstract: The EPSRC pilot project *Meeting the Design Challenges of NanoCMOS Electronics* (nanoCMOS) began in October 2006 and is focused upon tackling the decreasing scale of modern semiconductor components. This scaling has direct impact upon the complete circuit and system electronics design process due to the variability in transistor behaviour caused by differences in atomic structure. To address these challenges, the project has focused upon large scale device simulations exploiting a wide variety of computational resources. This paper focuses on the approach adopted for managing the many hundreds of thousands of files being generated that are associated with these simulations. Specific challenges in achieving this are related to the fine grained security demanded in protecting intellectual property of data and metadata, and the seamless linkage of metadata associated with services. We believe that this data architecture has widespread applicability to many research areas.

Overview: This paper presents a novel approach for tracking and modelling data in distributed environments exploiting web-based solutions for identification/addressing, linkage, access and usage of distributed data. The approach has been developed to support the management of input and output files and associated meta-data used in nanoCMOS simulations; however, other types of resources could be managed with this model.

In this model, experimental data and associated meta-data are both considered as resources, where a resource is an abstraction of any system entity. The main resources of the nanoCMOS data-management system are the input and output data of simulation software, as well as metadata associated with these files. Independently of the mechanism used for accessing or representing the resource, each resource is itself identified with its own URI (Uniform Resource Identifier).

Each resource may have a representation, which is a document that can be transferred between components. To the client accessing a given resource *R*, the representation of *R* should provide enough information for the client to be able to navigate to other related resources. For example, if *RF* is a resource that models information about a file, which itself links to other resources RA_1, RA_2, ..., RA_n (for example annotation resources), then the representation of *RF* should provide sufficient information so that the client can discover RA_1, RA_2, ..., *R* and, in turn, fetch their representations.

To model files, the system provides specialised *FileRecord* resources. A *FileRecord* is a place-holder that contains a set of core elements, including the URL (location) of the file, and a set of arbitrary annotations, each of which are also modelled as individual resources. The URIs used to identify resources are http-prefixed, and the implementation of these resources is provided canonically via an HTTP server.

This service may be used by a number of clients. In particular, it can be hidden behind a portal using a client API implemented in Java. It can also be accessed directly from applications running on compute-grids so as to populate the *FileRecord* resources with annotations regarding the execution of the simulation. A given *FileRecord* (that is, a model for a file and a set of annotations), or even a given annotation, can subsequently be used throughout the system by making links using its URI. The sole use of URIs as identifiers also makes it possible to use these resources in a wider context e.g. blogging about a *FileRecord* or annotation.

The security of this service is handled on a per-resource basis. Since each resource that needs to be protected is associated with an access-control resource and since each annotation is a resource in its own right, this mechanism allows for fine-grained access control to be established and enforced. It also allows for coarse-grained policies whereby a group of resources may refer to the same access-control resource. The representations an access control resource provides and accepts depends on which authorisation mechanism is used. This system could build upon existing authorisation systems such as PERMIS.

The full paper will describe the nanoCMOS project in more detail and the various types of simulation underway and hence the science that is currently supported. We will also describe in more detail the design of the nanoCMOS data management system and demonstrate its usage in managing files stored in secure file systems include the Andrews File System (AFS) and its use of Kerberos security tokens and their linkage with associated metadata.