SC2003 Tutorial Proposal

Introduction to the Scalable Systems Software Suite

www.scidac.org/ScalableSystems

Contact: Al Geist ORNL gst@ornl.gov

Presented by the Scalable Systems Software Working Group

with representatives from ORNL, ANL, NCSA, Ames Lab, LBL, SNL, and PNNL

1. Abstract

This half-day tutorial will describe a national effort to fundamentally change the way high-end systems software is developed and distributed. This effort, called the Scalable Systems Software project, involves DOE national labs, NSF supercomputer centers, universities, and industry. This group is collectively defining standardized interfaces between system components, using a process similar to that used to define MPI. At the same time this group is producing a fully compliant suite of systems software and tools. This suite is designed to work with large clusters as well as the supercomputers found in large computer centers. Pieces of this suite are already finding their way into OSCAR, City, and other popular cluster management packages.

The Scalable Systems Software suite is expected to make it much easier and cost effective for supercomputer centers to adapt, update, and maintain its systems software.

A beta release of this suite will be given out to attendees of this tutorial. The tutorial will include presentations on what is in the suite, how to install it, configuration options, and a demonstration of its use.

2. Description

2.1 Tutorial Goals

The primary goal of this half-day tutorial is to present and get audience participation in a new approach to the architecture of systems software. The Scalable Systems Software Project currently is developing a component architecture for systems software, including such components as a batch scheduler, process manager, node state manager, system monitor, checkpoint/restart manager, accounting manager, queue manager, scheduler, meta-scheduler, and others. Draft XML interfaces for these components have been published and prototype implementations of each component have been developed and will be available at the tutorial. Advantages of this approach will be explained, including the ability to replace individual components and the additional capabilities and options they make available to system managers. We hope to garner input from attendees to refine the architecture. A secondary goal is to teach attendees how to obtain, install and use the collection of components in their current state. The current suite of prototype component implementations provides a usable systems management solution that is particularly open to customization and experimentation.

2.2 Targeted Audience

The target audience for this tutorial is anyone who is interested in more cost effective management of large clusters, anyone who is involved in the running of computer centers, or anyone that is interested in getting involved in an effort to standardize system software interfaces especially vendors and researchers involved in the development of system software components.

2.3 Content Level – 20% beginner, 60% intermediate, 20% advanced

2.4 Audience Prerequisites

To get the most from this tutorial, attendees should have some experience using today's system software components either to manage large clusters, or in a computer center with multiple computer resources, or both. The standardized interfaces will be presented in XML so basic knowledge of XML syntax will be useful but not required.

2.5 Relevance to SC2003 Attendees

System administrators and managers of large computer centers are facing a crisis. These scientific computing centers all use incompatible, ad hoc sets of systems tools, and these tools were not designed to scale to the multi-teraflop systems that are being installed in these centers today. One solution would be for each computer center to take their home-grown software and rewrite it to be scalable. But this would incur a tremendous duplication of effort and delay the availability of terascale computers for scientific discovery.

The purpose of the Scalable Systems Software project is to provide a much more timely and cost effective solution by pulling together representatives from the major computer centers and industry to collectively defining standardized interfaces between system components. At the same time this group is producing a fully compliant suite of systems software and tools that make the management of large systems more cost effective and robust.

Attendees benefit by getting early releases of this software suite and instruction on its capabilities and installation. They also benefit by learning how they can become involved in the standardization process and be a part of the effort to fundamentally change the way future high-end systems software is developed.

2.6 Brief Description of Tutorial

This half-day tutorial is divided into three sections. The tutorial will begin with an overview of the effort and the philosophies behind the software architecture. We will also discuss our choices for the division of the system into functional components and the choice of XML for inter-component interfaces. A key feature of the architecture is that it enables the extensibility of the system by facilitating integrators and end users to customize the system by modifying our code or by swapping out a component entirely. This flexibility allows integrators and end users a range of choices when customizing the suite to best fit their individual hardware and site needs.

The second section of the tutorial will present the standardized XML interfaces between components. Our interface specification provides a fully documented set of interfaces for each component. Thus someone wishing to build a custom component can easily match the interfaces and behavior of the stock component. We will discuss the interfaces for each of the key system components describing both the mechanics and the functionality for the service directory and event manager, resource management system, process management and monitoring, and cluster build and configuration system.

The final part of the tutorial will focus on the details of installing, configuring and testing our reference software implementation. The installation demonstration will include building the software from source and installing from pre-built rpms. In both cases the software prerequisites will be covered and other potential installation problems will be examined. Part of the installation process will examine the security features of the suite and discuss how sites might modify the security to fit their local needs. Many components in the system are quite flexible and allow for a great deal of site-specific configuration. The tutorial will cover the existing configuration options..

2.7 How you will ensure cohesive tutorial content

Being a half-day tutorial we limit the number of people actually making presentations to two. We also use simple measures such as using the same template and style for all slides and presenting the entire tutorial from one laptop.

3. Detailed Outline of the Tutorial

This half-day (3 hour) tutorial is divided into three parts

1. (30 minutes) Overall description of the standardization effort,
    a. Goals and history
    b. Who is involved,
    c. Design philosophies – flexible, extensible, multi-language, multi-protocol
    d. How to join the effort or give input into the Scalable Systems standards process without joining.
    e. Questions and Answers about project

2. (60 minutes) Description of the XML Interfaces
    a. Resource and queue management,
    b. Accounting and allocation management,
    c. System and job monitoring,
    d. System build and configuration tools,
    e. Event manager and service directory.
    f. Questions and Answers about the standardization

3. (90 minutes) Description and use of the existing beta suite
    a. What is in it – components, and their functionality
    b. How to install it - prerequisites, process, and demonstration
    c. Demo its use – show components interacting and running jobs
    d. Questions and Answers about the systems software suite

4. Bios of the Presenters

Al Geist Oak Ridge National Laboratory gst@ornl.gov

Al Geist is a Senior Research Scientist at Oak Ridge National Laboratory (ORNL), where he has worked for 20 years. Al leads the Computer Science Research Group. He is one of the developers of the PVM software system, and participated in the design of both MPI-1 and MPI-2. Today He leads the national Scalable Systems Software effort, and one of five national Genomes to Life centers. He has participated in numerous popular SC tutorials in the past dealing with PVM and more recently Beowulf clusters. He has a wide range of interests and has published over 170 papers in solar energy, materials science, biology, solid state physics, numerical linear algebra, parallel computing, scientific computation, and distributed computing.

You can find out more about Al on his web page:
http://www.csm.ornl.gov/~geist

Brett Bode Ames Lab brett@scl.ameslab.gov

Brett Bode has been a Research Staff Member of Ames Laboratory since 1999 where he works on the development of high-performance computing solutions for scientific applications. He is a member of the Scalable Systems Software effort and is primary developer of a prototype queue manager component. He received his PhD in December 1998 in Theoretical Chemistry from Iowa State University. Brett has presented demonstrations of the Scalable Systems Software at SC2002 as well as at the 2003 SciDAC PI meeting.

Resume or Curriculum Vitae for each presenter (4 presenters maximum, 2-pages maximum each). Make sure this includes a list of short courses the presenter has taught.