The Next Generation Software Program

Frederica Darema
Computer and Information Science and Engineering Directorate
National Science Foundation
Arlington, VA 22230
darema@nsf.gov

Held in conjunction with IPDPS 2004 – 26 April – Eldorado Hotel, Santa Fe

Abstract

This workshop provides a forum for an overview, project presentations, and discussion of the research fostered and funded by the NSF Next Generation Software (NGS) Program[1]. The program announced in October of 1998, has had several calls for proposals (in FY99, FY01, FY02, FY03, and FY04), and supports research in two broad technical thrusts: One is in developing Technology for Performance Engineered Systems (TPES) for the Design, Management and Runtime Support of Computing Systems and Applications. The other thrust (Complex Application Development Support Systems - CADSS) seeks to create new systems' software technology, including enhanced compiler capabilities, and tools for the development, runtime support and dynamic composition of complex applications executing on complex computing platforms, such as Computational Grids, assemblies of embedded systems and sensor systems, as well as high-end platforms (Grids-in-a-Box) and special purpose processing systems.

1 Introduction

The program has had several calls for proposals (in FY99, FY01, FY02, FY03, and FY04), has received altogether over 350 submissions and has funded until now about 50 projects, ranging in duration from 2-5 years. The majority of the projects supported in the program involve multidisciplinary research, spanning several sub-areas in computer sciences. In addition the program puts emphasis in driving and validating with end-user applications the advances in Computer Sciences technologies.

The Technology for Performance Engineered Systems (TPES) component is aimed to enable systematic methods for the Design, Management and Runtime Support of Computing Systems and Applications, and emphasizing a system level approach for such analysis and prediction. To enable that, the program supports development of systematic methodologies and tools for analysis and prediction of the performance of applications, of hardware platforms, and of system software. The approach fostered by the NGS program is to consider the computing system as viewed in terms of an architectural framework, consisting of the applications, the system software and the underlying hardware. Key elements of the methodologies emphasized by the NGS program, include development of multi-level and multi-modal methods and tools for describing the application software, the system software, and the system hardware. Such models and tools encompass modeling and simulation of components at multiple levels of detail and abstraction, as well as incorporation of performance measurements. Additional key capability is the ability to combine these multi-level/multi-modal methods and tools into "performance frameworks" (in a "plug-and-play" fashion) as needed for understanding, analysis, and prediction of behavior and performance of individual components or layers as well as the system as a whole. Thus the methodology fostered by the NGS program enables component-level and system-level performance analysis and prediction.

The Complex Application Development Support Systems (CADSS) component of the program fosters new research in two key

[1] The NSF Next Generation Software (NGS) Program was developed and is managed by F. Darema. The Program Announcement and the list of current awards can be found at: http://www.nsf.gov/cgi-bin/getpub?nsf00134, and under the Next Generation Software (NGS) Program in http://www.nsf.gov/home/grants awards.htm.

technology areas: new compiling technology and new application composition technology, which will enable applications to effectively execute under dynamic runtime resource availability as is manifested in complex and heterogeneous Grid computing environments. In the new compiling system technology advocated here, part of the compiler gets embedded in the runtime, and a tight interaction will be enabled between the compiler and the underlying system resource managers as well as measurements, and performance descriptors of the applications and the underlying hardware and software systems. This kind of new compiling system (the Runtime Compiling System – RCS) will have the ability to adaptively optimize the mapping of applications on the underlying dynamic platform assembly. The NGS program also fosters the development of novel programming models technology, in conjunction of imparting to the RCS technology the desired capabilities.

In addition the program advocates the need for new technology for knowledge-based, RCS compiler-invoked dynamic assembly of application components capabilities, and supports research along these directions. NGS also fosters research for the integration of these technologies into application support environments, and demonstration of the technologies on important production-class applications. The presentations in this workshop provide a sample of the kind of research projects that are advancing technology along the directions envisioned in the NGS program.

2 Workshop Agenda

The NGS Workshop at IPDPS2004, features a several projects funded by the NGS program, and representing efforts sponsored under the TPES and the CADDS components. The presentations are listed below; the name underlined is that of the Principal Investigator of the NGS sponsored Project.

8:30 - 10:10 - **Session 1**

Introduction: The NGS Program
Frederica Darema

A Plan to Achieve Structural and Composable Modeling of Complex Systems
August, Malik, Peh, Pai

P2
Ciardo, et. al

Performance Modeling and Programming Environments for Petaflops Computers and the Blue Gene Machine
Zheng, Wilmarth, Lawlor, Kale, S. Adve, Padua, Guebelle

10:10-10:30 Break

10:30 - 12:00 - **Session 2**

The Weaves Runtime Framework
Varadarajan

Improvements in the Efficient Composition of Applications Built Using a Component-Based Programming Environment
Eidson, Eijkhout, Dongarra

Generic Programming and High-Performance Libraries
Jarvi, Lumsdaine, Gregor, Kulkarni, Musser, Schupp

Discussion

12:00 - 1:30 Lunch

1:30 - 4:00 - **Session 3**

New Grid Scheduling and Rescheduling Methods in the GrADS Project
Cooper, Dasgupta, Kennedy, Koebel, Mandal, Marin, Mazina, Mellor-Crummey, Berman, Casanova, Chien, Dail, X. Liu, Olugbile, Siever, Xia, Johnsson, B. Liu, Patel, Reed, Deng, Mendes, Shi, YarKhan, Dongarra

Autonomic Proactive Runtime Partitioning Strategies for SAMR Applications
Zhang, Yang, Hariri, Chandra, Parashar

Overhead Reduction Techniques for Software Dynamic Translation
Scott, Kumar, Childers, Davidson, Soffa

A Case Study Using Empirical Optimization for a Large Enginnerign Application
Diniz, Lee, Hall, Lucas

P11
S1, S2 Adve

Language and Compiler Design for Streaming Applications
Thies, Gordon, Karczmarek, Lin, Maze, Rabbah, Amarasinghe

4:00 - 4:30 Break

4:30 - 6:30 - **Session 4**

Liquid Architecture
Jones, Padmanabhan, Rymarz, Maschmeyer, Schuehler, Lockwood, Cytron

CMDL: A Class-based Machine Description Language for Co-Generation of Compilers and Simulators
Moss, Palmer, Richards, Walters, Weems

Hardware Support for Boundless Transactions
Ananian, Asanovic, Kuszmaul, Leiserson, Lie
& *Adversarial Analyses of Window Backoff Strategies for Simple Multiple-Access Channels*
Bender, Farach-Colton, He, Kuszmaul, Leiserson

Discussion