The NSF
Next Generation Software Program

Performance Engineering Technology

Dr. Frederica Darema
Senior Science and Technology Advisor
Director, Next Generation Software Program
NSF
Next Generation Software (NGS) Research Program

NGS Scope: performance methodologies, composition, and runtime support of complex applications with end-to-end performance, on adaptive, distributed heterogeneous platforms, including Grids, GiBs & embedded sensor systems

Program Components:
- TPES (Technology for Performance Engineered Systems)
- CADSS (Complex Application Development and Support Systems)
  - TPES: Methodology for performance engineered computing systems and applications
    - performance frameworks, layered and multilevel models for analysis and prediction of performance
      (DARPA BAA in ’96 on TPES)
  - CADSS: new software technology for application development and adaptive run-time support
    - new dynamic resource and performance-aware compiler/run-time
    - application composition technology
Application Directions

Past
- Mostly monolithic
- Mostly one programming language
- Computation Intensive
- Batch
- Hours/days

Present / Future
- Multi-Modular
- Multi-Language
- Multiple Developers
- Multi-source Data
- Computation Intensive
- Data Intensive
- Real Time
- Few Minutes/hours
- Visualization (real time)
- Interactive Steering
- Integrated Experiments & Simulations (Dynamic Data Driven Application Systems)

change at faster pace

+ Darema 11/10/2003
Platform Directions

- Vector Processors
- SIMD MPPs
- Distributed Memory MPs
- Shared Memory MPs

- Distributed Platforms, Heterogeneous Computers and Networks
  - Heterogeneity
    - architecture (computer, network)
    - node power (supernodes, PCs)
  - Latencies
    - variable (internode, intranode)
  - Bandwidths
    - different for various links
    - different based on traffic

Past

Vector Processors
SIMD MPPs

Present & Future

Distributed Platforms, Heterogeneous Computers and Networks

GiBs

Grids

Petaflops Platform (Grid in a Box)
Such systems cannot be designed, built, and supported in the ad-hoc ways of the past!

Need more systematic methods where system-level performance, QoS, dependability and fault tolerance are well understood and hopefully guaranteed.
Performance Engineering Technology (TPES) in the NGS Program
Systems Software/Hardware Architectural Framework

Applications/Users

Languages
Compilers
Libraries
Tools

Other Services . . .

Operating System

Distributed, Heterogeneous, Dynamic Computing Platforms and Networks

Application

API & Runtime Services

Global Management Computing Engine

Components Technology

Memory Technology

CPU Technology

Device Technology

Darema 11/10/2003
The NGS Program supports Research on Performance Engineering Technology (TPES) Frameworks, Models & Measurements

Application
APIs & Runtime Services
Global Management
Computing Engine
Components Technology

Performance Engineering Frameworks

Application Models

...

IO / File Models

OS Scheduler Models

Architecture / Network Models

Memory Models

Distributed Applications

Programming Env’s
Compilers
Libraries
Tools

Visualization
Collaboration Environments
Data Management
Archiving/Retrieval Services
Authentication/Authorization
Reliability Services

Other Services . . .

Distributed Systems Management

Distributed, Heterogeneous, Dynamic, Adaptive Computing Platforms and Networks

Memory Technology
CPU Technology
Device Technology

...
Enables Analysis in Multiple views of the system (The applications’ view)
Multiple views of the system
The Operating Systems’ view

- Distributed Applications
  - Visualization
  - Scalable I/O
  - Data Management
  - Archiving/Retrieval
  - Dependability
  - Authentication
  - Authorization
  - Other Services

- Distributed Systems Management

- Application Models
- IO / File Models
- OS Scheduler Models
- Architecture / Network Models
- Memory Models

- Languages
- Compilers
- Libraries
- Tools

- Distributed, Heterogeneous, Dynamic, Adaptive Computing Platforms and Networks

- Memory Technology
- CPU Technology
- Device Technology
Methodology

• consider the system in terms of its architectural layers and components
• consider multilevel, layered, multi-resolution and multi-modal approaches
  • combine different methods of describing components and layers
  • describe the system in multiple levels of detail (characteristics and time-scales)
• Performance Frameworks
  • combine tools in “plug-and-play” fashion
  • multiple views of the system
Innovations Needed for TPES

- Modeling languages for modeling and specification of performance attributes for such components and layers
  - applications, system software, hardware
- Methods of modeling and simulation at multiple levels of detail and abstraction, models/simulators embodying these methods
- Combine such multilevel and multimodal models and simulation tools (different levels of detail, different time-scales)
- Combine into the performance frameworks models and simulators of different resolution levels
- Measurement methods and tools
  - instrumentation methods to address heterogeneity (systems and time-scales)
  - integration of measurements with performance analysis tools
  - storage and retrieval of measurements and other performance data
- Integration of these technologies into performance frameworks
The CADDS component \textbf{Next Generation Software Program} develops Technology for \textit{integrated} feedback & control

\textit{Runtime Compiling System (RCS)} and Dynamic Application Composition

\textbf{CADSS}

- Application Model
- Application Program
- Application Intermediate Representation
- Application Components
- Distributed Programming Model
- Compiler Front-End
- Compiler Back-End
- Architecture & Performance Models/Measurements
- Distributed Computing Resources
- Distributed Platform
- MPP
- NOW
- SP

\textbf{Dynamic Analysis Situation}

Launch Application(s)

Dynamically Link & Execute

Adaptable computing Systems Infrastructure
Technical Areas

• **Application Programming System** *(dynamic compilation/run-time systems)*
  - distributed programming models for complex, distributed hardware platforms with complex memory structure and be adaptable to changes in the underlying platforms
  - compilers that interface with the system resource managers to request and determine resources, as well as interfacing with models of the underlying distributed hardware and software platforms to allow retargeting and optimizing application mappings on such complex systems
  - interfaces that allow applications to specify performance related parameters to enable applications to achieve quality of service

• **Application Composition System** *(dynamic selection of application components)*
  - technology for building knowledge-based systems allowing automatic selection of solution methods allowing applications to adapt to changes in the underlying platforms
  - application interfaces and methods for problem specification and extracting content information, standards of interfaces, data representation & exchange, and standard high-level and low-level libraries
  - interfaces to debugging tools and performance models

• **Performance Analysis System** *(technology for performance engineered applications)*
  - modeling languages and models for application and system description
  - interoperability of performance models of different levels of abstraction
  - methods and tools for measurement and instrumentation

• **Validation, Integration and Demonstrations**
  - validation of key technologies developed under each of the thrusts above
  - integration of the technologies developed above
  - demonstration of the ability of these technologies for design and runtime support of key applications executing under dynamically changing conditions
NGS Technology Roadmap

Application Programming System
- Distributed programming models
- Application performance interfaces
- Compilers optimizing mappings on complex systems

Application Composition System
- Automatic selection of solution methods
- Interfaces, data representation & exchange
- Debugging tools

Application Analysis System
- Application/system multi-resolution models
- Modeling languages
- Measurement and instrumentation

Providing enhanced capabilities for applications
Compiler/Runtime Systems

Application Programming & Composition Systems

Performance Engineered Systems
FY99 Awards

Advanced Compiling/Runtime Technology Projects
- Grid Application Development Software (GrADS) (Ken Kennedy-Rice University)
- An Integrated Framework for Performance Engineering and Resource-Aware Compiling System (Constantine Polychronopoulos – Univ. of Illinois)
- SmartApps: Smart Applications for Heterogeneous Computing (Lawrence Rauchwerger - Texas A&M)
- Compiling for Speculative Distributed Microarchitectures (Rudolf Eigenmann-Purdue University)
- A Staged Compilation Architecture for Program Optimization (Susan Eggers - U. of Washington)
- A Computing Environment for Adaptively Parallel Multithreading (Charles Leiserson, MIT)
- Active Debugging Information for Multi-Language, Multi-Platform Debugging (Norman Ramsey, Harvard)

Application Programming and Composition Projects
- Logistical QoS Through Application-driven Scheduling of Remote Storage (James S. Plank-Univ of Tennesse)
- TMO Based Modeling & Design of Reliable Next-Generation Complex Software (Kane Kim - Univ of California-Irvine)
- A Simulation Platform for Experimentation and Evaluation of Distributed-Computing Systems (SPEED-CS) (Suvrajeet Sen- University of Arizona)
- Adaptive, Performance-Portable Software for Next-Generation and Immersive Applications (Kai Li -Princeton University)
- Supporting Complex Application Requirements in Metasystems (Andrew Grimshaw - UVA)

Performance Engineering Projects:
- Model-based Management of Adaptive Programs on the Computational Grid (James Browne-University of Texas)
- Prophecy: A Performance Modeling Framework for the Analysis of Complex Applications and Systems (Valerie E. Taylor-Northwestern University)
- Coordinated Allocation of Processor and I/O Resources in Parallel Systems (Evgenia Smirni, College of William&Mary)
FY01 Awards

Compiler/Runtime Systems

Application Dynamic Composition Projects
- Computational Vortals for Next generation Scalable Computing (Tomasz Haupt - MissStateU.)
- Cache Efficient and Parallel Householder Diagonalization (Gary Howell - Fla Inst of Tech)
- Middleware for Internet Workflow Management (Marinescu – UCF)
- A Microarray Experiment Management System (Ramakrishnan, et al - Vtech)
- A Distributed Component Repository for Rapid Synthesis Adaptive Real-time Systems (Ling-I Yen, UT/Dallas)

Performance Engineering Projects:
- Distributed Optimizing Virtual Environment (DOVE) (Rastilav Bodik - UWisc/Madison)
- Performance Modeling and Programming environments for PetaFlop Computers and the Blue Gene Machine (Laxmi Kale - UIUC)
- POWERful Software for Power Constrained Systems (Sivasubramanian, et al - PennState)
- A Computer System Design Hierarchy for Simulation (Donald Thomas - CMU)
- Performance Driven Adaptive Software design and Control (Mary Vernon - UWisc/Madison; et al)
- Performance Mining of Large-Scale Data-Intensive Distributed Object Applications (Mohamed Zaki - RPI)

Advanced Compiling/Runtime Technology Projects
- CAREER: Techniques and Applications of Dynamic Compilations (Vikram Adve- UIUC)
- CAREER: Enhanced Dynamic Process management for Beowulf Clusters on the Grid (Daniel Andreesen - Kansas State U.)
- An OpenMP Environment for Wide-Area Network Computing (Eigenmann - Purdue; Padua - UIUC)
- A Framework for Developing Complex Applications on High-End PetaFlop-calls Machines (Guang Gao- Udelaware; et al)
- Efficient Script-Based Applications Development for Network High Performance Computing Environments (Ken Kennedy-Rice University ; et al)
- Open Compilation for self-Optimizing Generic Components (Andrew Lumsdaine - Indiana U)
- System and Compiler Support for Component-Based Construction of Scalable Internet Services (Thu Nguyen et al, Rutgers U.)
- SmartApps: An Application Centric Approach to High Performance Computing (Rauschwerger- TAM; Torellas - UIUC)

Application Programming & Composition Systems

Performance Engineered Systems
Advanced Compiling/Runtime Technology Projects:

• An Integrated Middleware and Language/Compiler Framework for Data Intensive Applications in a Grid Environment (Agrawal – OhioState)
• Continuous Compilation: A New Approach to Aggressive and (Davidson-UVA; Soffa - UPitt)
• Efficient Adaptabel Software via Staged Compilation (Chambers – U.Washington)
• Software Innovations for Liquid Architecture (Cyrton – WashU.SLouis)
• Resource-Aware Off-Line and On-Line Empirical Optimization (Hall – USC/ISI)
• An Application Development Environment for Complex Heterogeneous Distributed Real-Time Embedded Computing (Nicolau – UC Irvine)
• Agent Oriented Approaches to a Ubiquitous Grid (Joshi – UMD)
• Optimizing Performance and Reliability in Distributed Computing Systems through Wide Spectrum of Storage Services (Plank - U.Tennessee)
• Supporting Compiler/Simulator Co-Evolution for Architectural Exploration and Evaluation (Moss – UMassAmherst)
• CAREER: An In-kernel Runtime Execution Environment for User-Level Programs (Zadok – SUNYStonybrook)

Performance Engineering Projects:

• Structured Methods to Evaluate the Performance of Distributed Software (Ciardo – Coll W&M)
• End-to-End Performance Modeling of Applications Executing in the Internet2 Domain (Dickens – IIT)
• A Model Based Framework for Adaptive Algorithm Design (Prasanna – USC)

Application Dynamic Composition Projects:

• Mesh Generation and Optimistic Computation on the Grid (Chrisochoides – College W&M)
• Component-Based Frameworks for High-Performance Distributed Scientific Applications (Dongarra – U.Tennessee)
• Legion-G: Delivering a Scalable and Secure Programming Model for Grid Computing (Humphrey – UVA)
• Scalable Real-Time Simulation of Embedded Systems and Environments (Kim – UC Irvine)
• A Framework for Dynamic Composition and Reconfiguration of QoS Aware Nest Generation Software Systems (Menasce – George Mason U.)
• Towards Community Services: Putting Parallel Network Services On-line (Weissman – U.Minnesota)
Application Dynamic Composition Projects
- An Autonomic Component Framework for Grid Applications (Parashar – Rutgers; Hanri-UofArizona)
- A Framework for Dynamic Runtime Service on the Grid (Weissman-UofMinn)

Performance Engineering Projects:
- Structural and Composable Performance Simulation of Complex Systems (August-Princeton)
- A Component-Based Software Environment for Simulation, Emulation and Synthesis of Network Protocols in Next Generation Software (Hou-UlUC; Yao-Purdue; Yau-Purdue)
- Montage: An Integrated End-to-End Design and Development Framework for WirelessNetworks (Rappaport-UTAustin)
- Models to Support Performance Engineering of Global Computations (Wolski-UCSB)

Advanced Compiling/Runtime Technology Projects
- StreamIt: A Language and Compiler for Streaming Applications (Amarasinghe – MIT)
- Adapting Program Code Continuously and Aggressively (Davidson-UVA; Soffa - UPitt)
- A Computing Environment Based on Transactional Memory (Leiserson - MIT)
NGS Program Activities

• NSF NGS Program Announced in 1998 (started TPES w DARPA BAA in '96)
• FY99 (request: $112M; 77proposals)
  - TPES ($2.6M); CADSS ($8.6M; $4.7M)
• FY01 (request: $41M; 50 proposals)
  - TPES($5.15); CADSS ($5.7; $1.4)
• FY02 (request: $39M; 49 proposals)
  - TPES ($4M); CADSS ($3M; $2.9M)
• FY03($34M; 42 proposals); TPES:$3.1M; CADDNS ($3M)
• Additional ~NGS projects funded through the ITR Program (~ $2M in FY03 ITR/Small & ~$5.7M in FY03 ITR/Medium)
• Representative sample of NGS awards presented at NGS Workshops IPDPS'01, '02 and '03
• CADSS PIs meeting in Sept 2001 - Report
• TPES PIs meeting in Feb2002 - Report
Funded projects - Thematic Areas

• **Performance Engineering Projects (FY99):**
  - An Integrated Framework for Performance Engineering and Resource-Aware Compilation (PI: Polychronopoulos); includes Performance Engineering Framework (Bill Sanders)
  - Model-based Management of Adaptive Programs on the Computational Grid (James Browne-University of Texas)
  - Prophesy: A Performance Modeling Framework for the Analysis of Complex Applications and Systems (Valerie E. Taylor-Northwestern University)
  - Coordinated Allocation of Processor and I/O Resources in Parallel Systems (Evgenia Smirni, College of William&Mary)
Funded projects - Thematic Areas

• Performance Engineering Projects (FY01):
  • Distributed Optimizing Virtual Environment (DOVE) (Rastilav Bodik - UWisc/Madison)
  • Performance Modeling and Programming Environments for PetaFlop Computers and the Blue Gene Machine (Laxmi Kale - UIUC)
  • Adaptive Performance and Power Management for Real-Time Systems (Israel Koren - UMass/Amherst)
  • POWERful Software for Power Constrained Systems (Sivasubramanian, et al - PennState)
  • A Computer System Design Hierarchy for Simulation (Donald Thomas - CMU)
  • Performance Driven Adaptive Software Design and Control (Mary Vernon - UWisc/Madison; et al)
  • Performance Mining of Large-Scale Data-Intensive Distributed Object Applications (Mohamed Zaki - RPI)
Funded projects - Thematic Areas

• **Performance Engineering Projects (FY02):**
  • Structured Methods to Evaluate the Performance of Distributed Software (Ciardo – Coll W&M)
  • End-to-End Performance Modeling of Applications Executing in the Internet2 Domain (Dickens – IIT)
  • Supporting Compiler/Simulator Co-Evolution for Architectural Exploration and Evaluation (Moss and Weems – Umass/Amherst)
  • A Model Based Framework for Adaptive Algorithm Design (Prasanna – USC)
• Performance Engineering Projects (FY03):
  • Structural and Composable Performance Simulation of Complex Systems (August-Princeton)
  • A Component-Based Software Environment for Simulation, Emulation and Synthesis of Network Protocols in Next Generation Software (Hou-UIUC; Yao-Purdue; Yau-Purdue)
  • Montage: An Integrated End-to-End Design and Development Framework for Wireless Networks (Rappaport-UTAustin)
  • Models to Support Performance Engineering of Global Computations (Wolski-UCSB)
NGS webpage:


www.cise.nsf.gov/acir
What is DDDAS

OLD
(serialized and static)

NEW PARADIGM
(Dynamic Data-Driven Application Systems)

Theory
(First Principles)

Simulations
(Math Modeling
Phenomenology)

Experiment
Measurements
Field-Data
User

Challenges:
Application/Simulations Development
Application Algorithms
Computing/Runtime Systems Support
Enabling DDDAS

NGS Program

Performance Engineering

Dynamic Compilers & Application Composition

Dynamic Data-Driven Application Systems

Symbiotic Measurement & Simulation Systems
Enabling DDDAS

DDDAS Program will support multidisciplinary projects (Applications/Algorithms/CS Research).

Dynamic Data-Driven Application Systems -- Symbiotic Measurement & Simulation Systems

Dynamic Compilers & Application Composition

Performance Engineering

NGS Program supports computer science research.
www.cise.nsf.gov/dddas

www.dddas.org