

NSF DDDAS Workshop

Systems Software Working Group

M. Parashar, A. Sussman, J. Saltz

Notes from breakout on 01/19/06

WG4 - Working Group on Systems Software

- What is the state-of-the-art and what advances are needed in computer systems software methods and tools, and what new capabilities should be provided by the underlying systems and platforms on which these applications execute, so that quality of service is ensured?
- What are the software challenges in the programming environments for the development and runtime support, under conditions where the underlying resources as well as the applications requirements might be changing at execution time?
- What are the challenges to integrate real-time sensor and other measurement devices with special purpose data processing systems together with the parts of the application that execute in larger platforms and drive a seamless integration of stationary and mobile devices together with large high-end platforms, entailing grids that go beyond the present computational grids?
- What are the issues with respect to data management, data models and structures, and interfaces between simulations and measurements?
- What are the additional capabilities that are needed in the application support and systems management services?

Application Characteristics/Requirements

- Data acquisition
 - *sensor networks, data archives, instruments*
- Adaptations
 - *parameters, formulations, model, algorithms, workflows, implementation, ..*
- Definition of interfaces and data
 - *metadata specifications, ontologies*
- Resilience, fault-tolerance, survivability
 - *Checkpoint/restart*
 - *Dealing with uncertainty*
- Resource discovery, management
 - *access to current high-end resources*
- Computational engines
 - *adaptations, parallelism*
- Predictable/controllable response times

System Software Requirements

- Programming abstractions, systems
 - *programming adaptivity*
 - *specify what, when and how*
 - *specify “control” behaviors, policies and constraints*
 - *actuation (correctness, consistency, conflicts)*
 - *programming sensor systems*
 - *discovery, querying, in-network processing, resource management*
 - *runtime system middleware for dynamic execution*
 - *dynamic discovery, configuration, instantiation, invocation*
 - *system integration*
 - *integrating system components*
 - *dynamic compositions, interactions, coordination*
 - *dynamic, long-lived workflows*
 - *addressing legacy systems*
 - *“autonomics”*
 - *dealing with uncertainty*
 - *algorithms, middleware services*

System Software Requirements

- Data acquisition, management, integration
 - *high-level data management abstractions*
 - *metadata specification*
 - *address data distributions*
 - *define/evaluate data quality*
 - *“real” time data access, transport, analysis*
 - *data integration from multiple sources*
 - *data conversion*
 - *data volumes*
 - *acquisition of sensor data*

System Software Requirements

- Testbed (Framework)
 - *Community testbed to verify, validate, evaluate DDDAS applications*
 - *access issues for high-end resources for DDDAS*
 - *Virtualization*
 - *Software repository*
 - *Best practices*

System Software Requirements

- Other thoughts
 - *what is the “software stack” for DDDAS applications*
 - *horizontal v/s vertically integrated*
 - *separate call for enabling technologies?*
 - *different metric for productivity for DDDAS*
 - *time to science*

Discussion

- licensing
- scalability – transition from small scale to large scale
- software quality, assurance
- security
- add data sources (synthetic, ns2, etc.) as part of the testbed
- adding repository of reference versions
- separate workshop for software systems
- Grids!
- software stack – opportunity for leveraging Grid environments
- integrate computation, data, measurements, actuation Grids
- interact with other program – cyberinfrastructure

WG4 Breakout – 01/20/06

- **DDDAS Software Stack**
 - *programming models/systems*
 - *runtime environment*
 - *system service*
 - *miscellaneous*
- **DDDAS Testbed**
 - *hardware, software, data*
 - *benchmarks/workloads*
 - *models/simulations*
 - *integration with other testbeds*

DDDAS Software Stack

- Programming models/systems
 - *support for expressing and enforcing dynamic adaptivity*
 - *adapt components, interactions, mechanisms, compositions, workflows, deployments*
 - *system dynamics and application dynamics*
 - *sense states and discover options*
 - *queryable specification of hooks (sensors, actuators)*
 - *element runtime state, capabilities, adaptability*
 - *evaluate quality of data/information*
 - *policy/constraints specification and enforcement*
 - *negotiation and coordinated adaptations*
 - *support for long running applications*
 - *checkpoint/restart*
 - *application-level/system-level*
 - *survivability, fault tolerance, process migration*
 - *descriptive metadata specifications – quantitative/qualitative metadata for elements*

DDDAS Software Stack

- Runtime environment/middleware
 - *in-network data processing, assimilation*
 - *dealing with uncertainty, missing data*
 - *dynamic resource management and instantiation*
 - *support for checkpoint/restart*
 - *deal with “real-time” event, constraints*
 - *data transport, streaming, coupling*
 - *timing, format constraints*
 - *policy injection, enforcement, conflict detection and resolution*
 - *interfacing and virtualizing data, computation, instrument, monitoring Grids*

DDDAS Software Stack

- System services
 - *dynamic resource allocation/co-allocation*
 - *monitoring/event logging/accountability*
 - *service/data quality*
 - *time-synchronization service*
 - *content-based messaging/interaction/coordination*
 - *event triggers, notification*
 - *traceability*
 - *data/information virtualization*
 - *monitoring/probing*
 - *security, trust*

DDDAS Software Stack

- **Miscellaneous**
 - *logic system for describing, reasoning about, validating adaptive behaviors*
 - *debugging DDDAS applications*
 - *oscillations due to adaptations (infinite loops)*
 - *human in/out of the loop*
 - *software architectures*
 - *leverage real-time systems research, control theory, agent system research*
 - *predictability*
 - *integration of different capabilities, levels of abstraction, etc.*
 - *knowledge generation*

DDDAS Testbed

- *hardware/software components*
- *distributed individual resource at sites that connected to form a testbed*
 - *administrative issues*
- *distributed, virtualized*
- *leveraging existing efforts – Geni*
- *integrate simulators and models*
 - *data model – sensors sources, synthetic data, data streams, archives*
 - *sensor network system simulators*
- *integrate real sensor/instrument deployments*
- *benchmarks*
- *workloads*
- *end-to-end DDDAS applications*
- *usage model, interface*

Related programs (CS)

- cyberinfrastructure
- nets – sensors
- csr – aes
- data integration/ai
- nmi
- seiii