Dependable Computing Clouds for Cyber-Physical Systems

Dependability Issues in Cloud Computing (DISCCO) Workshop
October 11th, 2012

Douglas C. Schmidt
d.schmidt@vanderbilt.edu

Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN

http://www.dre.vanderbilt.edu/~schmidt/DCC4CPS.pdf
Outline of Presentation

• Context & terminology
• Prior R&D progress
• Current R&D challenges
• A promising solution
• Upcoming event
• Concluding remarks
Outline of Presentation

• Context & terminology

• Prior R&D progress

• Current R&D trends & challenges

• A promising solution

• Upcoming event

• Concluding remarks
Overview of Cyber-Physical Systems

- A cyber-physical system (CPS) features a tight combination of—and coordination between—the system’s computational & physical elements.
- A CPS often involves networked processing elements that control physical, chemical, or biological processes or devices.
- In a CPS the “right answer” delivered too late becomes the “wrong answer.”
- i.e., dependability has a temporal dimension (& increasingly a security dimension)

CPSs are increasingly distributed systems rather than standalone devices
Overview of Cloud Computing

• Cloud computing involves the delivery of computing as a service rather than as a product
  - Essentially a form of “outsourcing” for certain hardware & software components & activities

• Cloud offerings enable economies of scale via multi-tenancy & elasticity
  - Typically run atop shared (& virtualized) hardware, storage, data access, middleware, software, etc.

• Cloud services don’t require end-user knowledge of the physical location & configuration of the computing/comm infrastructure delivering the services
  - Similar to utilities, such as power grids, datacom/telecom, water, sewer, etc.

Key characteristics of clouds for CPS are multi-tenancy & elasticity
Outline of Presentation

• Context & terminology
• Prior R&D progress
• Current R&D trends & challenges
• A promising solution
• Upcoming event
• Concluding remarks
The designs of legacy CPSs tend to be:

- Stovepiped
- Proprietary
- Brittle & non-adaptive
- Expensive to develop & evolve
- Vulnerable

**Problem:** Small changes can break nearly anything & everything.
Prior R&D Progress for Cyber-Physical Systems

...and this operational paradigm...

Real-time QoS requirements for legacy CPSs:

- Ensure predictable end-to-end QoS, e.g.,
  - Minimize latency, jitter, & footprint
  - Bound priority inversions
- Allocate & manage resources statically & avoid sharing

Problem: Lack of any resource can break nearly everything
Prior R&D Progress for Cyber-Physical Systems

…and this operational paradigm…

Real-time QoS requirements for legacy CPSs:

- Ensure *predictable* end-to-end QoS, e.g.,
  - Minimize latency, jitter, & footprint
  - Bound priority inversions
- Allocate & manage resources statically & avoid sharing

This is not at all what we think of as a computing cloud!
Prior R&D Progress for Cyber-Physical Systems

...to this design paradigm...

The designs of today’s leading-edge CPSs tend to be more:

- Layered & componentized
- Standard & COTS
- Robust to failures & adaptive to operating conditions
- Cost effective to evolve & retarget

Result: changing requirements & environments can be handled more flexibly
Prior R&D Progress for Cyber-Physical Systems

…and this operational paradigm…

- Ensure acceptable end-to-end QoS, e.g.,
  - Minimize latency, jitter, & footprint
  - Minimize priority inversions
- Resources are allocated/managed dynamically & can be shared

Result: better support for operations with scarce/contended resources
Prior R&D Progress for Cyber-Physical Systems

...and this operational paradigm...

• Ensure acceptable end-to-end QoS, e.g.,
  • Minimize latency, jitter, & footprint
  • Minimize priority inversions
  • Resources are allocated/managed dynamically & can be shared

Some of these CPS computing environments have much in common with clouds
New Challenge: Ultra-Large-Scale Cyber-Physical Systems

Key problem space challenges
- Limited communication bandwidth
- Connectivity is intermittent & connections are noisy
- Processing & storage capacity are limited & power/weight limits affect usage patterns
- Must adapt to unanticipated workflows
- Frequent dynamic network topology & membership changes

Key solution space challenges
- Enormous accidental & inherent complexities
- Continuous evolution & change
- Highly heterogeneous platform, language, & tool environments

Mapping problem space requirements to solution space artifacts is very hard!
Cyber-Physical Systems-of-Systems well beyond the scope of clouds today

Key problem space challenges
- Limited communication bandwidth, connectivity is intermittent & connections are noisy
- Processing & storage capacity are limited & power/weight limits affect usage patterns
- Must adapt to unanticipated workflows
- Frequent dynamic network topology & membership changes

New Challenge: Ultra-Large-Scale Cyber-Physical Systems

Key solution space challenges
- Enormous accidental & inherent complexities
- Continuous evolution & change
- Highly heterogeneous platform, language, & tool environments
Outline of Presentation

• Context & terminology
• Prior R&D progress
• Current R&D trends & challenges
• A promising solution
• Upcoming event
• Concluding remarks
Convenient Trend: Elastic Hardware Platforms

- “Elastic hardware” based on multi-core & distributed-core architectures now available at reasonable prices
- Elastic hardware has potential to substantially accelerate performance by parallelizing application workloads & auto-scaling data processing at runtime
  - Goal is to add/utilize more hardware without changing application business logic or configurations
- Current focus of elastic hardware is largely on web hosting applications in public cloud environments

Elastic hardware is necessary—but not sufficient—for elastic CPS applications
Impediments to Elastic Hardware for CPSs

• Inadequate programming models
  – Complicated & obtrusive APIs
  – Can’t use hardware effectively & scalably

• Inadequate knowledge of real-time, concurrency, & networking
  – e.g., high probability of race conditions, deadlocks, priority inversion, & missed deadlines

• Inadequate mechanisms to transition seamlessly from multi- to distributed-core environments

• Inadequate at scale quality-of-service (QoS) support
  – e.g., lack of control over key QoS impacting resource usage & end-to-end data deliver semantics

Some impediments affect many types of systems, some mostly affect CPSs
Key Research Challenges for Elastic CPSs

- Precise auto-scaling of resources with a system-wide focus
- Flexible optimization algorithms to balance real-time constraints with cost & other goals
- Improved fault-tolerance fail-over that supports real-time requirements
- Data provisioning & load balancing algorithms that use physical properties of computations

These challenges are hard enough for non-elastic cyber-physical systems!
Key Research Challenges for Elastic CPSs

- Precise auto-scaling of resources with a system-wide focus
  - Lack of predictable workload patterns makes it hard to inform cloud providers of resource requirements
  - Current state-of-the-art in auto-scaling algorithms manage services in isolation
- CPSs require auto-scaling algorithms to operate on end-to-end task chains
  - Physical stability & safety properties may require exceedingly complex analyses
    - e.g., reachability of hybrid cyber-physical states
Key Research Challenges for Elastic CPSs

- Flexible optimization algorithms to balance real-time constraints with cost & other goals
  - CPS deployments must be schedulable on all resources acquired from cloud providers to ensure real-time response times while optimizing desired objective functions
    - e.g., minimizing costs
  - Principled means are needed to co-schedule or perform admission control & eviction of mixed-criticality task sets deployed on cloud resources

Multi-dimensional Resource Management

- ISR Processing
- SCADA Systems
- Air Traffic Mgmt
- Aerospace

Multi-core Chips
Symmetric Multiprocessors
Blade Clusters
Public/Private Clouds
Improved fault-tolerance fail-over that supports real-time requirements

- Although some cloud platforms tolerate faults for provisioned resources, this is insufficient for CPSs where real-time fault-tolerance of end-to-end task chains must be met simultaneously while minimizing costs.
- The complex & stochastic nature of some CPSs means that reasoning about consequences of faults is an important open research area.
Key Research Challenges for Elastic CPSs

- Data provisioning & load balancing algorithms that use physical properties of computations
  - CPSs generate load on a cloud computing environment due to physical stimuli
  - To build most scalable & high-performance systems, algorithms & techniques are needed to exploit physical characteristics of data & computation to improve the distribution of work in a cloud environment
Outline of Presentation

• Context & terminology
• Prior R&D progress
• Current R&D trends & challenges
• A promising solution
• Upcoming event
• Concluding remarks

We need a holistic solution that provides an elastic CPS software infrastructure
Requirements for Elastic CPS Software Infrastructure

- **Flexibility** – loosely coupled components that can be analyzed, replaced, reused, distributed, & parallelized dependably

- **Adaptability** – provide APIs that adapt to existing code, rather than always having to adapt code to an API

- **Uniformity** – seamless (ideally standards-based) support for multi-core & distributed-core

- **Scalability** – static & dynamic load balancing ensures best & dependable utilization of available elastic hardware resources

*Middleware is at the heart of elastic CPS software infrastructure*
Key Layers of CPS Software Infrastructure

Provide mechanisms to manage endsystem resources, e.g., CPU scheduling, file systems, memory management, & IPC

Domain-Specific Services
Common Middleware Services
Distribution Middleware
Host Infrastructure Middleware
Operating Systems & Protocols

VxWorks
Microsoft Windows
Encapsulates & enhances native OS mechanisms to create reusable network programming components.
Defines higher-level distributed programming models whose reusable APIs & components automate & extend native OS capabilities.
Augment distribution middleware by defining higher-level domain-independent services that focus on programming “business logic”
Key Layers of CPS Software Infrastructure

Tailored to the requirements of particular domains, such as SCADA, C4ISR, avionics, vehtronics, air traffic management, aerospace, etc.
Promising Elastic CPS Middleware: DDS

- The OMG Data Distribution Service (DDS) promotes a pattern language that leads to loosely coupled, polyglot, evolvable, scalable, high performance, & dependable, CPSs
  - DDS supports flat, relational, & OO information modeling
  - DDS global data space allows apps to read/write data anonymously/asynchronously, decoupled in space & time
  - DDS QoS-enabled pub/sub model allows apps to produce/consume information into/from the global data space

DDS mainly provides distribution middleware & common middleware services
Promising Elastic CPS Middleware: DDS

- DDS provides a rich set of QoS policies that control resource usage, end-to-end data delivery, & data availability, e.g.:
  - Batching
  - Priority
  - Deadline
  - Data Durability
  - Redundancy
  - Data History

en.wikipedia.org/wiki/Data_Distribution_Service has a good DDS overview
Promising Elastic CPS Middleware: DDS

- DDS provides a rich set of QoS policies that control resource usage, end-to-end data delivery, & data availability, e.g.:
  - Batching
  - Priority
  - Deadline
  - Data Durability
  - Redundancy
  - Data History

- Bridges are available across technologies to expose relevant data to over 80 communication protocols, without imposing changes into existing systems.

www.slideshare.net/Angelo.Corsaro/cloud-messaging-with-opensplice-dds
Promising Elastic CPS Middleware: DDS

- DDS is an OMG standard that itself is based on many associated open standards

- All mainstream DDS implementations are now available in open-source form

- DDS is used in many CPS research projects & production systems

- [http://portals.omg.org/dds](http://portals.omg.org/dds) provides more info on DDS open-source activities & projects
Outline of Presentation

• Context & terminology
• Prior R&D progress
• Current R&D trends & challenges
• A promising solution
• Upcoming event
• Concluding remarks
Upcoming NSF Workshop on “CC4CPS”

- www.isis.vanderbilt.edu/workshops/cc4cps contains information on an upcoming NSF workshop on Computing Clouds for Cyber-Physical Systems (CC4CPS)
  - Held March 14 & 15th, 2012 in Arlington, VA
  - Short position papers are due by November 30th, 2012
  - NSF will pay for ~$1k of travel expenses for each participant
Outline of Presentation

• Context & terminology
• Prior R&D progress
• Current R&D trends & challenges
• A promising solution
• Upcoming event
• Concluding remarks
Concluding Remarks

• Despite promising advances in elastic hardware, deploying CPSs in cloud environments is hard without adequate support from elastic software infrastructure
  - It’s unlikely that public clouds will work for mission-critical (ultra-large-scale) cyber-physical systems

• The standards-based DDS middleware ecosystem provides key open-source building-blocks for creating a dependable elastic software infrastructure for CPSs

“Big breakthroughs often happen when what is suddenly possible meets what is desperately necessary”
- Thomas Friedman
Concluding Remarks

- Despite promising advances in elastic hardware, deploying CPSs in cloud environments is hard without adequate support from elastic software infrastructure
  - It’s unlikely that public clouds will work for mission-critical (ultra-large-scale) cyber-physical systems
- The standards-based DDS middleware ecosystem provides key open-source building-blocks for creating a dependable elastic software infrastructure for CPSs

“Big breakthroughs often happen when what is suddenly possible meets what is desperately necessary”
– Thomas Friedman
Ultra-large-scale (ULS) systems are socio-technical ecosystems comprised of software-reliant systems, people, policies, cultures, & economics that have unprecedented scale:

- # of software & hardware elements
- # of connections & interdependencies
- # of computational elements
- # of purposes & perception of purposes
- # of routine processes & “emergent behaviors”
- # of (overlapping) policy domains & enforceable mechanisms
- # of people involved in some way
- Amount of data stored, accessed, & manipulated
- … etc …

See blog.sei.cmu.edu for more discussions of software R&D activities
The report focuses on ensuring the DoD has the technical capacity & workforce to design, produce, assure, & evolve innovative software-reliant systems in a predictable manner, while effectively managing risk, cost, schedule, & complexity.