Management Strategies for Software Infrastructure in Large-Scale Cyber-Physical Systems for the US Navy

by Nick Guertin, Brian Womble, Paul Bruhns, Douglas C. Schmidt, Adam Porter, and Bill Antypas

For over a decade, the US Navy has been modernizing many of its large-scale, mission-/safety-critical, and software-intensive national security systems using an *open systems architecture* (OSA)[1] approach, which leverages capable and reliable standards-based commercial off-the-shelf (COTS) software infrastructure components and modern software development practices for iterative and incremental development. As with other mission-/safety-critical domains, such as air traffic management, power grid, and automotive systems, a key goal of the Navy's OSA strategy has been to field and manage affordable and superior capabilities more rapidly at reduced cost.

Although Navy OSA efforts have yielded some success, new challenges have arisen in the context of large-scale *cyber-physical systems* (CPSs), which play a critical role in existing and planned naval combat systems that combine computers, networks, and sensors to track and engage with enemy threats. This article describes the needs of — and advances in — software infrastructure management that are emerging to support the requirements of large-scale CPSs for the US Navy, as well as commercial systems with similar requirements, including the Internet of Things (IoT) and Industrial Internet.

CURRENT PROGRESS IN MANAGING INFRASTRUCTURE FOR SOFTWARE-INTENSIVE NAVAL SYSTEMS

As the result of collaborations between government, industry, and researchers over many years, the US Navy's national security systems and IT system upgrades are now routinely accomplished using COTS components and modern software development practices, demonstrating that the US Navy has achieved measureable success in managing infrastructure for software-intensive systems. For example, the Navy's Consolidated Afloat Networks and Enterprise Services (CANES)[2] infrastructure provides a common computing and communication platform, including standardized network protocols, operating systems, middleware, and user interfaces. The CANES platform enables:

- Continual hardware and software component upgrades via an adaptable COTS-based infrastructure that can evolve quickly to meet changing warfighting requirements
- Reduction in total ownership costs via consolidation of support functions and open competition throughout acquisition program lifecycles

Other examples of recent progress in managing infrastructure in OSA-based IT systems are the US Navy's Maintenance Free Operating Period (MFOP)[3] pilot projects. MFOP is defined as the specified period of time that a system must be available in support of its required mission, with a specified level of reliability and with little/no maintenance of hardware or software platforms.[4] Several MFOP pilot projects were conducted on Navy surface and subsurface ships over the past decade to determine the feasibility of (1) providing greater than 99% probability for completing a deployment on a combat ship of 180 days with no open cabinet maintenance, and (2) eliminating as much of the traditional shipboard maintenance support package as possible.

Shipboard support packages have classically been developed using such tools as the failure modes, effects, and criticality analysis (FMECA) method. These approaches stock spare components of a system based on a risk tolerance of mission success in the face of hardware failure, mean-time logistics delay time, and the like. For most fielded fleet systems, 100% component sparing is not affordable or necessary. With an MFOP design, the number of needed spares and associated maintenance support can be greatly reduced. This includes onboard repair parts, technical manuals for diagnostics and repair, trained maintenance personnel, and the shore-based infrastructure needed to sustain them. The cost benefits of these design practices include a reduction in capital outlay for spare hardware, as well as reduction in the cost to develop unique training curricula, training facilities, training faculty, and time commitments of the operational crew.

In these MFOP pilot projects, the Navy used hardware redundancy and software management to ensure the continuous availability of a capability (e.g., situational awareness for deploying marine expeditionary forces or passive sonar detection for submarines) needed at sea for a six-month or longer deployment period. In the most recent surface combatant MFOP pilot program, a blade center was configured with a pool of standby hardware assets, such as redundant compute nodes, separated disk storage, SCSI interfaces to RAID 6 storage area networks (SANs), separate and redundant power supplies, redundant Internet protocol (IP) switches, and redundant interfaces to other shipboard systems. The blade center provided firmware-based control modules that monitored all the system's hardware functions. The control modules, in turn, were remotely reachable by IP networks using a Web-based interface that enabled operators to monitor and manually control the hardware remotely. The hardware redundancy in the MFOP pilot projects supported the overall availability requirements, while the software and operator interface components played an essential role in ensuring responsiveness.

For example, in the MFOP pilot projects, agents in the compute nodes were used to monitor performance. These agents reported status to an application called the Remote Off Hull Monitoring System (ROHMS), which stored report data from application servers. If a server encountered a problem or a compute node failed, the ROHMS would send an alert from the ship along with additional information about the potential source of the problem collected by the agent. A separate server running a blade-center fabric manager would simultaneously initiate a failover action plan prescribed by the system IT operator. The compute node's Logical Unit Numbers (LUNs) and World Wide Names (WWNs) would be reassigned automatically to a backup compute node that assumed responsibility for subsequent processing, and a seamless rollover to the SAN RAID 6 file system would initiate. This logical-to-physical reassignment dramatically reduced active-passive replication server response time to the order of several seconds, rather than minutes or longer in traditional system configurations.

EMERGING CHALLENGES AND ADVANCES IN MANAGING INFRASTRUCTURE FOR SOFTWARE-INTENSIVE NAVAL SYSTEMS

Existing Navy OSA efforts, such as CANES and the MFOP pilot projects we just described, have been largely successful. New challenges have arisen, however, that require continued innovation in R&D to support the needs of next-generation naval mission capabilities and requirements in the context of large-scale cyber-physical systems.[5] CPSs interconnect and coordinate physical and computational elements to control physical, chemical, or biological processes.

Large-scale CPSs in the naval domain must operate reliably over broad scales of distribution, resource consumption, and utilization to support mission-/safety-critical operations. There is a key temporal dimension to reliability in a CPS, since the right answer delivered at the wrong time becomes the wrong answer. For example, it does no good to compute the optimal strategy for evading an incoming torpedo if the answer arrives after the ship has been struck.

There is a key temporal dimension to reliability in a CPS, since the right answer delivered at the wrong time becomes the wrong answer.

Large-scale CPSs for the Navy continue to grow in size and complexity to address expanding threats, such as sea-skimming cruise missiles and theater ballistic missiles. They also continue to evolve to meet demands for rapid fielding of technology innovations. As a result, significant and novel requirements have arisen for new types of software infrastructure management strategies that are not well supported by conventional information technologies and methods. In particular, strategies for managing software infrastructure (such as virtualization techniques applied in COTS cloud computing platforms) do not yet adequately address the challenges posed by the next generation of large-scale naval CPSs. These challenges include a combination of dynamic demand for resources and rigorous control over timing and physical properties, as well as functional properties, spanning diverse system scales and geographic distributions. Fulfilling these requirements motivates innovations that can achieve a new foundation for *computing clouds for cyber-physical systems* (CC4CPSs)[6] to meet the needs of large-scale CPSs. The remainder of this section describes how advances in CC4CPS software infrastructure management are needed to support the challenges and end-to-end requirements of large-scale CPSs for the US Navy.

1. Precise auto-scaling of resources with an end-to-end focus. In conventional cloud environments, auto-scaling adds cores dynamically when demand rises. For

example, when load increases, cloud services are provisioned with higher demand on existing resources and may be granted access to new resources, depending on service-level agreements (SLAs). Likewise, the cloud infrastructure can de-provision resources when load levels abate. Due to the lack of effective means to predict workload patterns in large-scale CPSs, however, it is hard to inform cloud providers of resource requirements, which means that auto-scaling may be complicated or even infeasible in practice. As a result, extra resources may be allocated (which is wasteful since it may degrade SLAs for other applications or services sharing the cloud platform), or insufficient resources may be allocated (which can adversely affect system deadlines and response times).

Although conventional cloud auto-scaling mechanisms are useful, they are not designed to support end-to-end resource management in large-scale CPSs.

Although conventional cloud auto-scaling mechanisms are useful, they are not designed to support end-to-end resource management in large-scale CPSs. For example, conventional auto-scaling algorithms in cloud infrastructures manage one service at a time in isolation. In contrast, large-scale CPS applications are composed of interacting services, so they require auto-scaling algorithms that operate at the level of service groups working together in end-to-end task chains, while ensuring that end-to-end cyber-physical quality-of-service (QoS) requirements are met. Management strategies for large-scale CPS infrastructure therefore require means for scaling up scheduling and auto-scaling in a broader environment; for example, to support precise behavior in end-to-end task chains for sets of mission capabilities with different criticality levels. Stability and safety properties within these mission-critical large-scale CPSs require complex analyses (such as reachability of hybrid cyber-physical states) that must be evaluated and enforced to provide confidence that they work as expected for both desirable and undesirable operational states in a large-scale CPS.

2. Optimization algorithms that balance real-time constraints with cost and other goals. Management strategies in conventional cloud environments primarily optimize performance by adding hardware. The size, weight, and power restrictions in naval deployments, however, often limit the computing and communication resources available to them. Likewise, the cost of acquiring and sustaining these resources across a program's lifecycle can be prohibitively expensive. Although deployment and configuration algorithms — along with services and infrastructure — are key to successful large-scale CPSs, implementing these algorithms effectively is hard in domains that are cost-sensitive. For example, remotely piloted aircraft in the naval aviation domain often cannot afford thousands of dollars' worth of high-end hardware because the resulting solution would be prohibitively expensive relative to mission needs and budgetary constraints.

Since large-scale CPSs are realized as end-to-end real-time task chains, their deployment on cloud resources must be schedulable on all resources acquired from the cloud infrastructure to ensure real-time response times while optimizing desired objective functions, such as minimizing operational costs. These requirements must be met in the context of the enacted auto-scaling algorithms. Due to different criticalities of task chains that could be deployed on computing and communication resources, a verified means of co-scheduling or performing admission control and/or eviction of mixed-criticality task sets is also needed. We require new ways of reasoning about these issues, including algorithms that trade off flexibility (e.g., by provisioning additional resources in non-critical operating modes, such as video streaming for crew entertainment) with predictability (e.g., by ensuring timelines are met in mission- or safety-critical operating modes, such as air and missile defense).

3. Improved fault-tolerance failover that supports real-time requirements. This requirement is crucial in large-scale CPSs that incur high levels of cyber-attacks and failures due to external forces. A common means of supporting fault tolerance in conventional cloud environments is *passive replication*, where each request is processed on a primary replica and the resultant state is transferred to a secondary replica when a failure occurs. Passive replication, however, is insufficient to meet the real-time QoS requirements of large-scale CPSs. Conversely, purely *active replication* models are often highly complex and may consume excessive resources, which is problematic in naval deployments that are limited by size, weight, and power restrictions.

A promising technique for these types of CPSs is *semi-active replication*, which enables running systems to failover rapidly and predictably.[7] This replication style provides some benefits of both the active and passive replication styles, including predictable failover times and predictable behavior during program execution. In general, the complex and potentially stochastic nature

of large-scale CPSs means that even reasoning about the consequences of faults and the trajectories of system behaviors resulting from them is an open area of research.

4. Data provisioning and load balancing algorithms that can take into account a variety of properties. Large-scale CPSs generate load on a cloud computing environment due to physical stimuli, such as traffic, power grid fluctuations, human movement, and changing weather patterns. Conventional cloud computing environments are generally configured to maximize flexibility. As a result, there is little need to consider where computation takes place or where data is stored, which makes sense since there are no real-time QoS needs.

To meet the real-time QoS needs of large-scale CPSs, however, it is essential to consider exactly where parts of the system computations and data are located. In particular, deployment mechanisms and load-balancing algorithms in the software infrastructure must consider the physical characteristics of data and computation when distributing work in a CC4CPS environment. In this context, affinity-based task scheduling and placement (including geophysical position) are needed to reduce latency and jitter when deciding where to migrate work when overloads or failures occur. For example, it may be necessary to cluster data onto nodes based on geographic associations, social network linkages, or other physical-world aspects. Understanding the relationship between physical-world aspects and cyber optimizations to improve scalability and response time of cloud systems is critical to supporting CPSs.

THE CONNECTION BETWEEN LARGE-SCALE CPSs FOR THE US NAVY AND FOR THE COMMERCIAL WORLD

It is important to note the direct connection between the special characteristics that drive the design and management of large-scale CPS software infrastructures for the naval domain and those found in commercial domains, such as the power grid, telecom systems, and air traffic control systems. Systems with extreme safety or security concerns, such as nuclear reactor control systems or commercial flight avionics, also fall into this category. But beyond these obvious examples, the emerging IoT and Industrial Internet depend upon the same challenging management and control characteristics as Navy large-scale CPSs. The dimensions of scale (device-to-device or machine-to-machine), resource allocation, redundancy management, maintenance-free operation, semantic interoperability, portability, dynamic provisioning, regulatory conformance, and direct control of effectors and sensors are present equally in both Navy CPS and commercial IoT and Industrial Internet systems of systems. It is therefore likely that techniques developed to manage the infrastructure of Navy large-scale CPSs will be equally applicable to commercial domains.

CONCLUDING REMARKS

At first glance, advances in infrastructure management strategies for conventional cloud computing seem well-suited for naval CPSs since they both address large-scale deployment environments. In practice, however, the former have been optimized around costs and volume, whereas the latter must be optimized around real-time performance and dependability to meet their mission- and safety-critical requirements. In particular, IT business environments rarely require the microsecond response times and predictability that naval CPSs do. Moreover, rarely does the failure of an IT system imply a risk of life or limb for employees. In contrast, as naval systems continue to evolve and grow in complexity and capability, they are being integrated to create large-scale CPSs, where the right answer delivered at the wrong time becomes the wrong answer. Software infrastructure management strategies and technologies that meet the stringent requirements of these naval systems — as well as support their cyber security, modularity, and safety needs — are essential to resolve shortfalls in conventional cloud infrastructure and better address the needs of large-scale CPSs.

ENDNOTES

[1]"Open Systems Architecture (OSA)." United States Navy Fact File, updated 18 December 2013.

[2]"CANES – Consolidated Afloat Networks and Enterprise Services." Program Executive Office (PEO), US Navy.

[3]Guertin, Nickolas, and Paul Bruhns. "Comparing Acquisition Strategies: Maintenance-Free Operating Period vs. Traditional Logistics Support." *Proceedings of the 8th Annual Acquisition Research Symposium*. Naval Postgraduate School, 30 April 2011.

[4]"Maintenance-free operating period" (Wikipedia).

[5]"Cyber-Physical Systems – A Concept Map." UC Regents, 2012.

[6]Schmidt, Douglas C., Chris Gill, and Jules White. "Elastic Infrastructure to Support Computing Clouds for Large-Scale Cyber-Physical Systems." *Proceedings of the 17th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2014.

[7]Gokhale, Aniruddha, et al. "Towards Real-time Fault-Tolerant CORBA Middleware." *Cluster Computing*, Vol. 7, No. 4, October 2004.

Nickolas Guertin, P.E, is the Director for Transformation in the office of the Deputy Assistant Secretary of the Navy for Research, Development, Test, and Evaluation (DASN RDT&E). He has extensive experience in open systems architecture (OSA) product development for weapons, sensors, and ship systems, as well as expertise in ship construction and repair. Mr. Guertin leads the transformation of business, technical, and cultural practices for OSA acquisition of national security systems as a coordinated Naval Enterprise effort. He also leads the chartered Department of Defense (DoD) OSA and Data Rights Team supporting the Better Buying Power initiative. He can be reached at nickolas.h.guertin@navy.mil.

Brian Womble spent the first half of his career working in the telecommunications industry centered in Dallas, Texas, as a system developer. After relocating to northern Virginia, he worked as a Systems Architect for Cisco Systems on VoIP programs. Mr. Womble then joined Lockheed Martin to work on open architecture efforts within the US Navy Submarine Advanced Rapid COTS Insertion (ARCI) program. Mr. Womble joined the Navy civilian work force, and is now leading efforts to transition the Naval Enterprise to OSA and open business models. He can be reached at brian.womble@navy.mil.

Paul Bruhns supports the DASN RDT&E office for the Business Innovation Initiative and OSA. He has extensive experience developing and fielding US Navy submarine acoustic and tactical information systems. Mr. Bruhns served as an original team leader for the ARCI program office, translating fleet requirements to specifications that can be implemented using COTS hardware and open software standards. He recently led a successful project to demonstrate cross-program reuse of tactical software products as part of a fielded Maintenance-Free Operating Period (MFOP) demonstration for surface ships. He can be reached at paul.bruhns.ctr@navy.mil.

Douglas C. Schmidt is a Professor of Computer Science at Vanderbilt University. He is also an Adjunct Professor of Software Engineering in the Institute for Software Research at the School of Computer Science at Carnegie Mellon University and a Visiting Scientist at the Software Engineering Institute (SEI). Dr. Schmidt has published 11 books and more than 500 technical papers covering a range of software-related topics, including patterns, optimization techniques, and empirical analyses of object-oriented frameworks and domain-specific modeling environments that facilitate the development of middleware and mission-critical applications running over networks and embedded system interconnects. He can be reached at d.schmidt@vanderbilt.edu.

Adam Porter is a Professor of Computer Science at the University of Maryland (UMD). He is also a Visiting Scientist and Research Fellow at the SEI and the Fraunhofer Institute for Experimental Software Engineering. Dr. Porter's research focuses on developing tools and techniques for large-scale software development. He is a winner of the National Science Foundation Faculty Early Career Development Award and the Dean's Award for Teaching Excellence in UMD's College of Computer, Mathematics, and Physical Sciences. He can be reached at aporter@cs.umd.edu.

William Antypas, Jr. currently supports NAVAIR PMA 209 OSA initiatives. He has broad experience in academia, commercial industry, defense contracting, and government DoD programs. Dr. Antypas has been Chief Scientist or Lead Engineer for several large-scale open systems, including CANES, Global Hawk Advanced Mission System, and other autonomous vehicles. He is a principal author of the Future Airborne Computing Environment (FACE) standard and a contributor to others, including SAE AS-4UCS. He can be reached at antypaswg@crltechnologies.com.