Disruption-Aware Dynamic Component Deployment and Composition in Ultra-Large-Scale (ULS) Systems

Yuan Xue and Shanshan Jiang, Vanderbilt University

I. Challenges of ULS Systems

ULS systems [1] are characterized by hundreds of thousands of hardware platforms and software systems connected through hierarchies of heterogeneous wireline and wireless networks. The development and maintenance of ULS systems is extremely hard due to the decentralization, dynamics, and heterogeneity of the computation and communication infrastructures that support these systems. The criticality of ULS systems requires high assurance and resilience against a wide spectrum of disturbances, including failures of system parts, as well as physical and cyber attacks.

Component-based software development focuses on building large software systems by integrating previously-existing software components [2], [3]. At the foundation of this approach is the assumption that certain parts of software systems reappear with sufficient regularity that common parts (*i.e.*, the *components*) can be reused as the basis for assembling a ULS system. In theory, the flexibility and maintainability of component-based software can help reduce software development costs, enable fast system assembling, and reduce the maintenance burden for ULS systems. In practice, however, composing a ULS system from reusable components is problematic due to the following unresolved research challenges:

1) The *highly dynamic and unpredictable* behavior of ULS systems prevents the application of static reliability analysis. Existing research [4] on reliable component deployment assumes a static network setting where network topology, node and link reliability are fixed and known *a priori*. Since these assumptions are unrealistic for ULS systems, new reliability and availability analytical frameworks are needed to capture the traditional concepts of instantaneous robustness and the time-sequenced concept of robustness that arise in dynamic ULS systems.
2) ULS systems require *decentralized* component deployment and recovery algorithms that can scale up to hundreds of thousands of hardware platforms and software components. Existing algorithms either are based on centralized assumptions or require precise and/or global system information [5], [6], [7] to make decisions, which limit the scalability of these algorithms. New algorithms that operate on *partial, incomplete and imprecise* information are therefore needed to guide component deployment and recovery decisions in ULS systems.
3) ULS systems involve *heterogeneous* applications and users with different *subjective* needs wrt system quality of service (QoS), such as reliability and availability. Existing research primarily focuses on low-level system reliability metrics, such as normalized reliability of composed service graph [4], and neglects higher-level user-perceived QoS. New component deployment and recovery strategies are therefore needed to support these different groups of users and reflect subjective human elements.

II. Solution Approach → Disruption-Aware Dynamic Component Deployment and Composition

To address the ULS system challenges described above, we posit the need to enhance the foundations of ULS system availability analysis, design decentralized component deployment and recovery algorithms, and build adaptive middleware platforms to support high-resilience ULS system development and deployment. In particular, research is needed on the following topics:

- Defining metrics that quantitatively characterize the user-perceived system QoS and the impact of service disruption in ULS systems. These metrics are critical to direct and evaluate the design of component deployment and recovery algorithms. To characterize the impact of service disruption fully, the frequency, duration, and scope of disruptions must be defined precisely at multiple ULS system levels, *e.g.*, system-wide, end-to-end, and local. Utility functions then will be defined to reflect the (dis)satisfactions of various types of users towards service disruptions in a flexible way.

- Investigating analytical resilience models that account for key spatial and temporal factors that influence the availability computation complexity of ULS systems. Based on the defined availability metrics, these models will develop optimization-based formulations for component deployment and composition under system dynamics. The goal is to maximize the user-perceived system QoS and minimize the impact of service disruption. The optimization-based analytical models will also provide valuable theoretical insights to algorithm design based on nonlinear/dynamic programming [8].
- Developing decentralized algorithms that provide scalable solutions for component deployment, composition, and failure recovery. These algorithms need to approximate and adapt the theoretical models to scale to ULS systems environments. Examples include: (1) *hierarchical deployment algorithms* that decentralize the deployment process via clustering and make deployment decisions only based on intra-cluster and local information and (2) *incremental component re-deployment* algorithms that minimize the number of redeployed components and the service interruption while maintaining the QoS requirements of critical system execution paths.
- Building integrated runtime platforms to facilitate ULS system development and deployment at multiple distributed systems levels, including middleware, operating systems, and networks. An example of such a platform is the *Resource Allocation and Control Engine* (RACE) [9], which provides a framework for managing the use of various nodal system resources (such as network bandwidth, CPU, and memory) by selectively applying algorithms designed to meet application QoS requirements and operating conditions. RACE also separates resource allocation and control algorithms from the underlying middleware, operating system, and network platforms so that these algorithms can reuse common mechanisms to (re)configure and (re)deploy components properly onto nodes in a ULS system.

This talk will explore the characteristics of ULS systems that affect component deployment, composition, and failure recovery and will present the results of our initial attempts to develop and validate component (re)deployment and (re)composition strategies for ULS systems in highly dynamic and unreliable network environments. We will describe how these efforts help improve ULS system availability, minimize the impact of disturbances, and support high-quality system execution. We will also discuss gaps in the existing technology base that require significant R&D investment to scale up to meet the ambitious demands of the next generation of ULS systems.

III. Author Bios

Yuan Xue is an Assistant Professor of Computer Science at Vanderbilt University. Her research focuses on optimization and statistics-based techniques for resource management, quality of service and reliability support in next generation networking and distributed systems such as wireless and sensor networks, peer-to-peer and overlay networks, and multimedia systems.

Shanshan Jiang is a second year Ph.D. student of Computer Science at Vanderbilt University. She received her B.S. and M.S. in Computer Science from University of Science and Technology of China (USTC). Her research focuses on component deployment and configuration with quality of service and reliability support in large-scale distributed real-time and embedded (DRE) systems.

References

[1] S. E. Institute, *Ultra-Large-Scale Systems: The Software Challenge of the Future*, Pittsburgh, 2006.

[2] V. Subramonian, G. Deng, C. Gill, J. Balasubramanian, L.-J. Shen, W. Otte, D. C. Schmidt, A. Gokhale, and N. Wang, "The Design and Performance of Component Middleware for QoS-enabled Deployment and Conguration of DRE Systems," *Journal of Systems and Software: special Issue Component-Based Software Engineering of Trustworthy Embedded Systems*, 2006.

[3] G. Deng, J. Balasubramanian, W. Otte, D. C. Schmidt, and A. Gokhale, "DAnCE: A QoS-enabled Component Deployment and Conguration Engine," in *Proceedings of the 3rd Working Conference on Component Deployment*, Nov. 2005.

[4] M. Mikic-Rakic, S. Malek, and N. Medvidovic, "Improving Availability in Large, Distributed Component-Based Systems via Redeployment," in *Proceedings of the 3rd International Working Conference on Component Deployment*, Nov. 2005.

[5] X. Gu and K. Nahrstedt, "Dynamic QoS-Aware Multimedia Service Configuration in Ubiquitous Computing Environments," in *Proceedings of IEEE International Conference on Distributed Computing Systems*, 2002.

[6] D. Wichadakul, K. Nahrstedt, X. Gu, and D. Xu, "2K: An Integrated Approach of QoS Compilation and Reconfigurable, Component-Based Run-Time Middleware for the Unified QoS Management Framework," in *Proceedings of ACM Middleware*, 2001, pp. 373–394.

[7] D. Xu, D. Wichadakul, and K. Nahrstedt, "Multimedia Service Configuration and Reservation in Heterogeneous Environments," in *Proceedings of IEEE International Conference on Distributed Computing Systems*, 2000.

[8] Y. Cui, Y. Xue, and K. Nahrstedt, “Optimal Resource Allocation in Overlay Multicast,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 8, 2006.

[9] J. Balasubramanian, P. Lardieri, D. C. Schmidt, G. Thaker, A. Gokhale, and T. Damiano, “A Multi-layered Resource Management Framework for Dynamic Resource Management in Enterprise DRE Systems,” *Journal of Systems and Software: special issue on Dynamic Resource Management in Distributed Real-Time Systems*, 2006.