Overfitting Machine Learning for Autonomic Enterprise DRE Pub/Sub Systems

Joe Hoffert and Douglas Schmidt
Vanderbilt University, EECS Department, Nashville, TN
{jhoffert, schmidt}@dre.vanderbilt.edu

I. Introduction

Emerging trends and challenges. Enterprise distributed real-time and embedded (DRE) publish/subscribe (pub/sub) systems manage resources and data that are vital to the ongoing objectives of organizations or projects. Examples include shipboard computing environments, air traffic management systems, and recovery operations in the aftermath of regional or national disasters. These enterprise DRE pub/sub systems often adjust the way they operate depending on their external environment. For example, search and rescue missions as part of disaster recovery operations can adjust the image resolution used to detect and track survivors depending on the resources available (*e.g.*, computing power, network bandwidth) [5].

Many enterprise DRE pub/sub systems autonomically (1) monitor their environment and (2) modify their modes as the environment changes since manual adjustment is too slow and error prone. For example, a shift in network reliability can prompt quality of service (QoS)-enabled middleware, such as the OMG Data Distribution Service (DDS), to change mechanisms (such as the transport used to deliver data) since some transports provide better reliability than others in some environments. Likewise, cloud computing applications where dynamically allocated resources (*e.g.*, CPU speeds and memory) cannot be characterized accurately *a priori* may need to adjust to available resources (such as using compression algorithms optimized for given CPU power and memory) at system startup. If adjustments take too long the mission(s) the system implements could be jeopardized.

Policy-based approaches (which externalize and codify logic to determine the behavior of managed systems [2]) are commonly used for the autonomic adaptation of enterprise DRE pub/sub systems. These approaches provide deterministic response times to perform appropriate adjustments given changes in the environment and can be optimized to ensure efficient performance. The complexity of developing and maintaining policy-based approaches for enterprise DRE pub/sub systems can be unacceptably high, however, since developers must determine which policies are applicable for certain environmental aspects. Moreover, developers must manage how the policies interact to provide needed adjustments.

Machine learning techniques support algorithms that allow systems to adjust behavior based on empirical data, *e.g.*, inputs from the environment. These techniques can be used to support autonomic adaptation by learning appropriate adjustments to various operating environments. Unlike policy-based approaches, machine learning techniques automatically recognize complex sets of environment aspects and make appropriate decisions accordingly.

Conventional machine learning techniques, such as decision trees and reinforcement learning, have been used to address autonomic adaptation for non-DRE systems [3]. These techniques are not well-suited for enterprise DRE pub/sub systems, however, since they do not provide bounded times when determining adjustments. Some techniques, such as reinforcement learning, explore the solution space until an appropriate solution is found, regardless of the elapsed time [1]. Other techniques, such as decision trees, have time complexities that are dependent upon the specific data and cannot be determined *a priori*. Moreover, decision trees may contain branches that are much longer than others which is a source of jitter for determining appropriate adaptations—an undesirable quality in DRE pub/sub systems.

Solution approach → Overfitted machine learning to guide pub/sub middleware adaptation. The goal of machine learning is to provide guidance for past known environments and to handle new and unknown environments. This generality sacrifices some accuracy, however, that would otherwise be provided for known environments. Machine learning techniques that are specialized for the environments they have seen—and/or on which they have been trained—are said to be *overfitted*, which reduces development complexity and makes the accuracy comparable to policy-based approaches.

This paper describes an overfitted machine learning approach we tailored to reduce the complexity of developing autonomically adaptive enterprise DRE pub/sub systems. In particular, we are tuning an artificial neural network (ANN), which is a technique modeled on the interaction of neurons in the human brain [4], to retain as much information about specific environment configurations and adjustments as possible (*e.g.*, greatly increasing the number of connections between input environment characteristics and output adjustments that are typically used in an ANN). Our work integrates this learning technique with the DDS QoS-enabled pub/sub middleware.

II. Key Challenges of Enterprise DRE Pub/Sub Systems

Below we summarize key challenges that arise when developing autonomic enterprise DRE pub/sub systems.

A. Challenge 1: Reduction of Development Complexity

Developing autonomic behavior can incur high complexity due to the number and type of relevant environmental conditions. For example, the number of data receivers can affect the optimal transport protocols and parameter settings used since some protocols provide adequate QoS for a small number of receivers whereas other protocols provide adequate QoS for a larger number of receivers. Codifying this knowledge requires developers to manually manage multiple aspects of the operating environment simultaneously, which increases development complexity.

B. Challenge 2: Timely Adaptation to Dynamic Environments

Due to the dynamic environment inherent in enterprise DRE pub/sub systems, application operations (such as image compression to reduce network traffic or disseminating data with both timeliness and reliability properties) must adjust in a bounded timely—ideally constant time—manner as the environment changes. Operations that cannot adjust quickly and in a bounded amount of time will fail to perform adequately when resources change, *e.g.*, if resources are lost or withdrawn—or demand for information increases—operations must be configured to accommodate these changes with appropriate responsiveness to maintain a minimum level of service. If resources increase or demand decreases, operations should adjust as quickly as possible to provide higher fidelity or more expansive coverage. Manual modification is often too slow and error-prone to maintain QoS.

C. Challenge 3: Accurate Adaptation to Dynamic Environments

Application operations in enterprise DRE pub/sub systems must be able to adjust to changes in the environment accurately. As known changes in enterprise DRE pub/sub systems occur (*e.g.*, an increase in networking capability, requests for data from additional senders and receivers) the system should take advantage of additional resources or provide for access to additional data producers and consumers to maintain or increase QoS. For a given environment configuration, the enterprise DRE pub/sub system must accurately implement adjustments that are appropriate to the operating environment.

III. Solution Approach - Overfitting Machine Learning Techniques

Our solution approach overfits machine learning techniques to increase accuracy in determining appropriate adjustments, such as applicable image compression algorithms for systems whose resources cannot be accurately characterized *a priori*. This approach enables enterprise DRE pub/sub systems to autonomically adjust to their environments. Moreover, we are leveraging techniques that provide time complexity guarantees needed for enterprise DRE pub/sub systems.

Our approach focuses on tuning an ANN to retain a high degree of information about specific environment configurations and adjustments, *e.g.*, increasing the number of *hidden nodes* [1] used in an ANN. As the ANN learns, it strengthens or weakens the connections between inputs, hidden nodes, and outputs to provide appropriate adjustments. Increasing the number of hidden nodes increases the level of detail that the ANN maintains. Our approach resolves the challenges presented in Section II as follows:

- Overfitted machine learning techniques address Challenge 1 in Section II-A by decreasing the development complexity involved with codifying adjustments for multiple configurations of operating environments. Policy-based approaches for autonomic adaptation place the complexity burden on application developers, who must manually maintain operating environment configurations, appropriate adjustments needed, and the mapping between the configurations and the adjustments. Moreover, developers must accurately codify this mapping in their implementations. Overfitted machine learning techniques relieve developers of this burden since they manage the complexity via training to react appropriately, such as the appropriate image compression algorithm that best matches the CPU speed and memory provide.
- Machine learning techniques that utilize a static number of equations for learning address Challenge 2 in Section II-B by providing constant time complexities for determining appropriate adjustments. In particular, we are applying overfitted ANNs to QoS-enabled pub/sub middleware to support enterprise DRE pub/sub systems. This technique incorporates appropriate algorithm selection or transport protocol adjustments according to feedback provided while the technique is being trained. When a technique is used in an enterprise DRE pub/sub system, the time to determine an appropriate adjustment is bounded by the constant number of equations involved.
- Overfitting the machine learning technique addresses Challenge 3 in Section II-C by increasing the accuracy of the learning techniques. Our approach increases the accuracy of determining appropriate adjustments for specific operating environments by increasing the number and layers of hidden nodes that connect the operating environment aspects, such as CPU speed and network bandwidth, with the appropriate adjustments, such as applicable image compression algorithms. In addition, we use non-overfitted machine learning techniques to guide adjustments when operating environments have not been previously anticipated. In this context, we want generalized machine learning to provide configuration assistance whereas policy-based approaches provide none.

We are integrating OpenSplice DDS and OpenDDS with transport protocol frameworks whose protocols provide customized QoS for different environments. In particular, we provide a DDS "environment monitoring" topic as input to our machine learning algorithms to determine the most appropriate protocol given the environment and support the desired QoS.

[1] Hidden nodes are the computational components that provide connections between the relevant aspects of the operating environment (*e.g.*, CPU speed, network reliability) with the adjustments needed for those environments.

References

[1] X. Bu, J. Rao, and C.-Z. Xu. A reinforcement learning approach to online web systems auto-configuration. In *ICDCS '09: Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems*, pages 2–11, Washington, DC, USA, 2009. IEEE Computer Society.

[2] A. R. Choudhary. Policy based management in the global information grid. *International Journal of Internet Protocol Technology*, 3(1):72–80, 2008.

[3] A. Hess, M. Nussbaumer, H. Hlavacs, and K. A. Hummel. *Principles, Systems and Applications of IP Telecommunications. Services and Security for Next Generation Networks*, chapter Automatic Adaptation and Analysis of SIP Headers Using Decision Trees. Springer Berlin / Heidelberg, 2008.

[4] D. W. Patterson. *Artificial Neural Networks: Theory and Applications*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[5] N. Shankaran, X. Koutsoukos, C. Lu, D. C. Schmidt, and Y. Xue. Hierarchical control of multiple resources in distributed real-time and embedded systems. *Real-Time Systems*, April 2007.