

# An Approach using the Data Distribution Service as the Connecting Transport for 100X Joint Battlespace Infosphere Servers

Lei Zhao, Dr. Douglas Blough, Dr. Vincent J. Mooney III, and Justin Fiore  
School of Electrical and Computer Engineering and College of Computing  
Georgia Institute of Technology, Atlanta, GA, USA

## Introduction

The Joint Battlespace Infosphere (JBI) is an information management system which allows users to dynamically provide, discover, and exchange information ([1], [2], [3]) in a publish-subscribe manner. The Data Distribution Service (DDS) supports the publish-subscribe paradigm and provides 22 Quality of Service (QoS) attributes [4]. For example, the Reliability QoS policy tells DDS whether to use a best effort or guaranteed data delivery approach. A DDS/JBI hybrid would be able to leverage the QoS support of DDS. We explore an integration of the JBI with DDS as a means of injecting QoS features into the JBI. Specifically, we provide DDS-based, QoS-enabled communication between multiple JBI servers. Tests were conducted to measure and evaluate QoS support and the system's robustness. We focused on the 100X JBI implementation [1] of the JBI. The only known open source C++ implementation of DDS, TAO DDS, later renamed to OpenDDS [5], was chosen. The details of this work are available in a technical report [6].

## Architecture

One interesting feature of the 100X JBI is the 100X JBI Connector. The 100X JBI Connector code executes alongside 100X JBI servers and serves to connect multiple 100X JBI servers. Figure 1 shows an example of three 100X JBI servers and their 100X JBI Connectors. In implementing the candidate integration, DDS is utilized between JBI servers. The DDS code is transparent to the JBI servers and JBI clients, so that JBI clients interact with the JBI servers in exactly the same way as before. We call the code based on DDS the "DDS/JBI Connector."

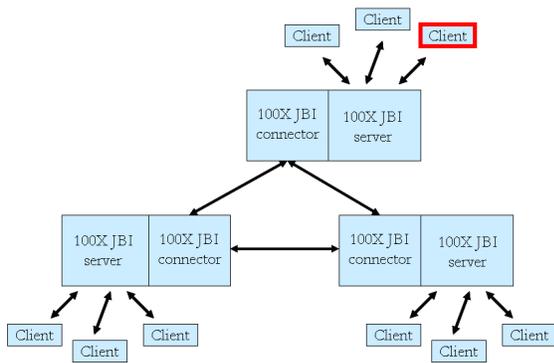


Figure 1: An example 100X JBI setup.

The main benefit of the DDS/JBI Connector is QoS-enabled dissemination between multiple JBI servers. The DDS/JBI Connector is also capable of compressing JBI IOs. In Figure 2, three DDS/JBI Connectors are shown linked in an all-to-all fashion. Note also the connections between the DDS/JBI Connectors and the DCPSInfoRepo (part of the TAO DDS architecture), shown as dashed lines. All JBI IO types are represented with just one DDS message type and, currently, one DDS topic.

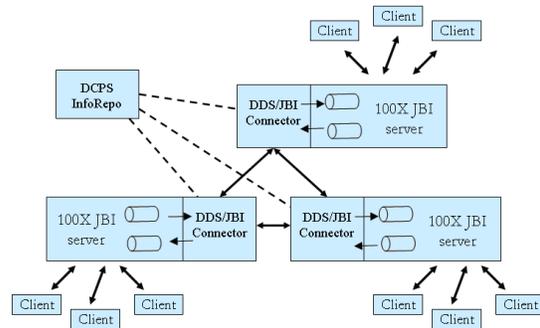


Figure 2: Integration details.

## Results

The DDS/JBI Connector has been demonstrated functionally in two node and three node configurations. TAO DDS currently supports five of the 22 QoS policies defined in the DDS specification [5].

*History/Durability:* These settings determine if samples are discarded by DDS after being sent to all known subscribers or if a certain number of samples is kept to send to late-joining subscribers. This operation was verified with a three-node configuration experiment and can be leveraged to allow a new Connector or a restarted Connector to get a snapshot of the past.

*Liveliness:* TAO DDS supports the Automatic setting, which indicates that DDS should periodically poll participants at a configurable interval. When an SSH tunnel is broken or a DDS/JBI Connector exits, the DDS components on the nodes are notified within the lease duration. Notification is also given to the DDS components when the DDS/JBI Connector which was previously not reachable is again reachable (the SSH tunnel is restored and/or the DDS/JBI Connector restarts). However, one major issue identified in the course of this investigation is that there is no

mechanism within the current JBI ([1], [3]) with which the DDS/JBI Connector can communicate to indicate liveness information. That is, the DDS/JBI Connector has no way of notifying the JBI that a node has gone down or reappeared.

*Reliability and Resource Limits:* Only the use of TCP transports with Reliability = Reliable ensured message delivery. Other settings resulted in samples being dropped in cases of heavy load for tests (or error message indicating that the maximum blocking time has been exceeded).

Performance results in terms of latency were obtained for the 100X Connector, the DDS/JBI Connector with compression, and the DDS/JBI Connector without compression for a message type of approximately one kilobyte in size. The results indicate that most of the latency can be attributed to network communication time for both the 100X and DDS/JBI Connector and not to the overhead of the connectors.

## Conclusion

We have implemented a candidate architecture for JBI-DDS integration. The work has demonstrated the establishment of improved QoS for transportation amongst multiple 100X JBI servers. We have identified a number of issues. The DDS/JBI Connector uses the pre-existing 100X JBI queues, which are First-In-First-Out (FIFO), JBI IO-type agnostic queues with no sense of priority. The implementation of DDS used supports only five QoS policies and has a single point of failure. At least one TAO DDS bug has been uncovered in the course of this work. Furthermore, QoS information – e.g., liveness – cannot be passed from the DDS components to the current JBI due to lack of QoS support in the current JBI [1], [3]. There are several technical challenges involved, and more work needs to be conducted to study the injection of QoS at a deeper level.

## References

- [1] L. Kwong-Yan, R. Linderman, and G. Ramseyer, “100X Joint Battlespace Infosphere (JBI),” AFRL/IFTC, Rome, NY, Tech. Rep. AFRL-IF-RS-TR-2006-265, 2006.
- [2] United States Air Force Scientific Advisory Board, “Report on Building the Joint Battlespace Infosphere, Vol. 1: Summary,” Tech. Rep. SAB-TR-99-02, 17 December 1999.
- [3] JBI Program Office, AFRL/IFSE, “Reference Implementation Quick Start Guide: Core Services Reference Implementation, Version 1.2.6,” 14 December 2005, [http://www.rl.af.mil/programs/jbi/pages/documents/files/1.2.6/RI\\_Quick\\_Start\\_Guide.pdf](http://www.rl.af.mil/programs/jbi/pages/documents/files/1.2.6/RI_Quick_Start_Guide.pdf).
- [4] Object Management Group, “Data Distribution Service for Real-Time Systems Specification v1.2,” 1 January 2007, <http://www.omg.org/docs/formal/07-01-01.pdf>.
- [5] Object Computing, Inc. “Data Distribution Service,” 9 February 2007, <http://download.ocweb.com/OpenDDS/previous-releases/DDS-0.12.pdf>.
- [6] L. Zhao, J. Fiore, V. Mooney, and D. Blough, “An Approach using the Data Distribution Service as the Connecting Transport for 100X Joint Battlespace Infosphere Servers,” Technical Report, 2007.