OBJECT COMPUTING

HOME TO THE THE OPENDDS® PROJECT

WEBINAR

Scalability and Performance Testing with the OpenDDS Bench Framework

Timothy Simpson
Principal Software Engineer

OpenDDS FOUNDATION™

OpenDDS Foundation
12140 Woodcrest Exec. Dr., Ste. 300
Saint Louis, MO 63141 USA

Limits of Liability and Disclaimer of Warranty
While every precaution has been taken in preparing this material, including research, development and testing, the OpenDDS Foundation assumes no responsibility for errors or omissions. No liability is assumed by the OpenDDS Foundation for any damages resulting from the use of this information.

AGENDA

- Introduction
- Motivation For Bench
- Current Capabilities of Bench (Part 1)
- Bench Architecture
- Configuring Bench
- Demo
- Bench Capabilities (Part 2)
- Future of Bench
- Q&A

Introduction

ABOUT ME

- Timothy Simpson
  - simpsont@objectcomputing.com
  - simpsont-oci on GitHub
- 15+ years performance-oriented C++ development
  - Remote Sensing, Defense, Financial Industries
  - Embedded & Real-Time Processing
- Principal Software Engineer @ Object Computing
  - 5+ years with OCI
  - 3+ years on the OpenDDS project

What is Bench?

OpenDDS "Bench" is a scalability and performance-testing framework for enabling large-scale testing of OpenDDS discovery and transport mechanisms

- Currently resides in the OpenDDS repository under `performance-tests/bench`
- Current documentation resides under `docs/internal/bench2.rst`
- Small test scenarios are run during OpenDDS CI process for GitHub pull requests
- Larger scenarios are run in an OCI test environment after each merge to master
    - Results are published to: http://scoreboard.ociweb.com/bench2

Motivation

Motivation for Bench (v2)

Previous Iteration of Bench:

- As seen on https://opendds.org/about/performance.html
- Able to:
  - Capture latency, jitter, and throughput values
  - Support a number of QoS settings and transport configurations
- Limitations:
  - Limited support for different network / participant topologies
  - No statistics for discovery times or machine statistics (e.g. CPU utilization)
  - “Heavy” wrapper of DDS (support for new configurations required new test code)
  - Active IO during tests (SSH connections to in tests, disk IO for writing raw statistic values)

Capabilities (Part 1)

Capabilities of Bench (v2)

- Parity with Bench v1
  - Support for the same basic statistics, scenarios, and outputs as Bench v1
- Ability to test OpenDDS correctness and performance at larger scales
  - Support for increasing number of machines and arbitrary network / DDS entity topologies
  - Support for arbitrary OpenDDS configurations without additional test code
- Better isolation of test performance
  - No test framework network IO activity during active test period
  - Test processes avoid non-OpenDDS memory allocations or disk IO during active test period
- Support for more kinds of statistics
  - Discovery times and "per-node" statistics (e.g., CPU and memory utilization)

Architecture

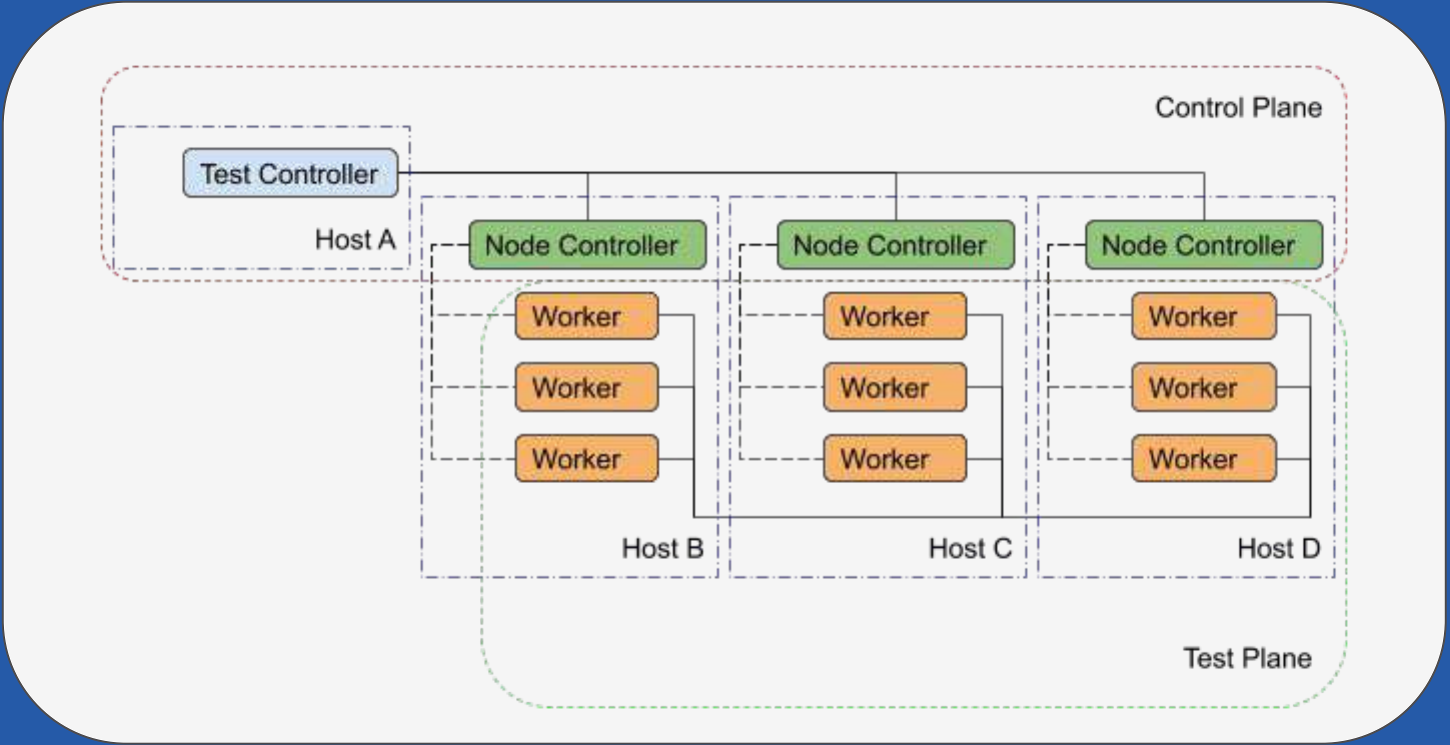
Bench Architecture: Three Process Design

- "Worker" Process:
  - Does most of the "test" work
  - Creates and destroys DDS entities
  - Writes and reads DDS sample data
  - Behavior determined by JSON config
- "Node Controller" Process:
  - Listens for requests to launch worker processes
  - Creates and destroys worker processes
  - Collects and reports worker outputs
- "Test Controller" Process:
  - Runs JSON scenario configuration files
    - Worker files and counts, node constraints
  - Allocates workers to available node controllers
  - Collects node controller and worker reports

Bench Architecture: Two “Plane” Design

- All three processes use DDS for communication
    - To isolate test performance data, DDS communication operates on two different planes
- Test Plane
    - Worker-to-worker test traffic
- Control Plane
    - Requests from Test Controller to Node Controllers
    - Reports from Node Controllers to Test Controller

Bench Architecture



Configuration

Worker Configuration Files

- Sections:
    - Test Timings (enable / start / stop / destruction)
    - Global Parameters
    - OpenDDS Configuration Sections (.ini equivalent)
    - DDS Entities (participants / topics / writers / etc)
    - Test “Actions”
        - Write New Sample Data (count / rate / size)
        - Forward Received Sample Data
        - Set Content Filtered Topic Parameters

Worker Configuration Files (Examples)

Echo "Client" Example

Echo "Server" Example

Scenario Configuration Files

- Sections:
  - Name / Description
  - Full-Node Descriptions
    - For when workers must run on the same node
    - Includes counts and “exclusivity”
  - Any-Node Descriptions
    - For when workers may run on any free node
    - Includes counts
  - Test Timeout

Scenario Configuration Files (Example)

Echo Test Scenario (using client & server from before)

Demo

Capabilities (Part 2)

Current Active Use of Bench

- OpenDDS Continuous Integration
  - Currently used for a number of light-weight stress tests on GitHub pull request workflows
- OpenDDS Performance Tracking
  - Merges to OpenDDS master branch are pulled down to an internal OCI test environment
  - A number of larger performance test scenarios are run for each merge commit
  - Results are post-processed and made available on our performance dashboard
  - Legacy (Bench v1 style) results are uploaded to performance web page
- Rapid Prototyping / Debugging
  - Theoretical Deployments: Checking for performance bottlenecks / modeling QoS
  - Bug Duplication: Reproduce client issues without client code
  - Easy Portability: Code available to clients, configuration files easily shared

Future Capabilities

Potential Future Capabilities of Bench

- Cleaner JSON Configuration Files, Support for Schemas
- Better Support for DDS Security Scenarios
- Statistics for Builtin Topics
- Additional Statistics (dropped packets, or alternative throughput calculations)
- Support for Easier Display of Time-Series Statistics
- Allow Multiple Test “Phases” For Simulating Topology Changes
- Support for User-Supplied Types (for testing Serializer and XTypes performance)
- Others: What would be of value to current and potential OpenDDS users?

Learn More About OpenDDS Support and Training Options

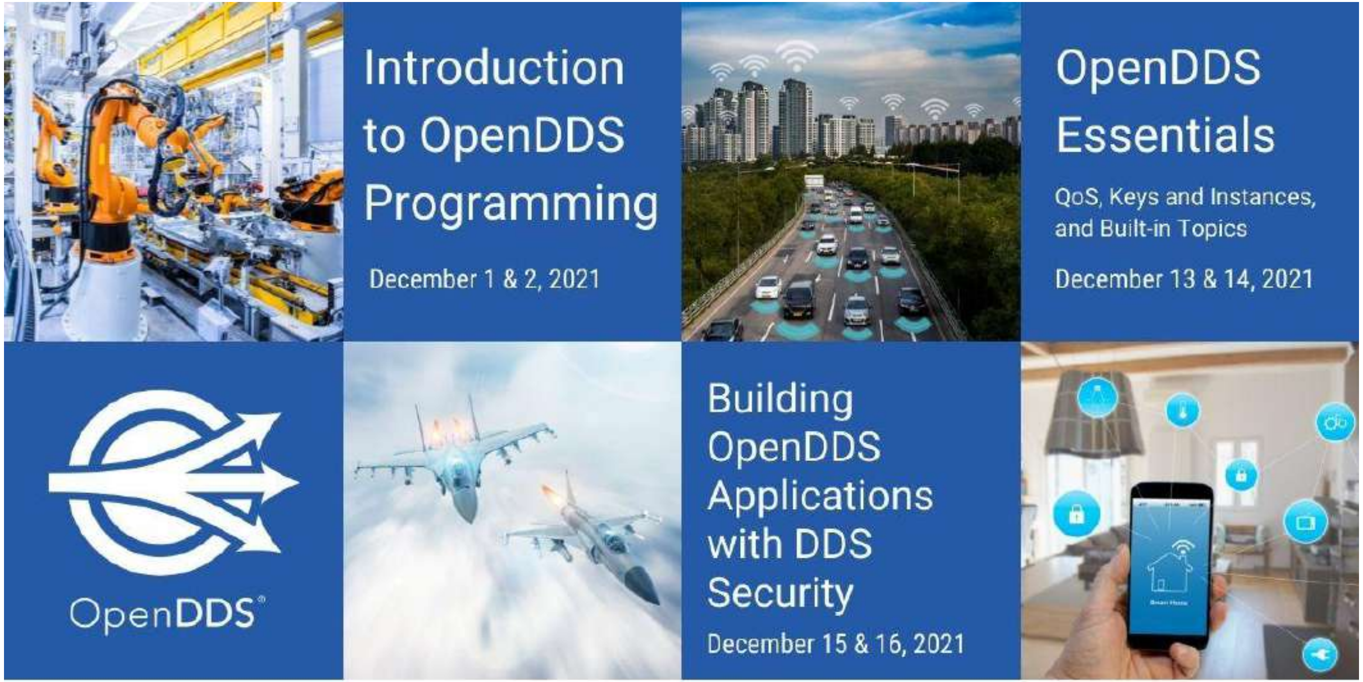
Plan, prioritize, and execute against your strategic digital initiatives with expert support services provided by the core OpenDDS development team.

- Custom training engagements (virtual or on-site)
- Architecture and design consulting
- Porting, extending, integrating
- Performance and scalability analysis
- Research and development
- ... and more!

Contact Object Computing to learn more: objectcomputing.com/opendds-consulting-and-support

WINTER 2021 CLASSES ENROLLING NOW!

objectcomputing.com/events



Introduction to OpenDDS Programming

December 1 & 2, 2021

OpenDDS Essentials

QoS, Keys and Instances, and Built-in Topics

December 13 & 14, 2021

Building OpenDDS Applications with DDS Security

December 15 & 16, 2021

OpenDDS FOUNDATION™

OpenDDS Foundation is a not-for-profit organization that exists to support and collectively lead the open source OpenDDS® project. The Foundation is supported by a Technology Advisory Board that ensures the technology continues to reflect and serve its diverse and growing user community.

OpenDDS Foundation works to ensure technical innovation and advancement of the OpenDDS project, evangelize and promote the project as a leading technology in the data distribution space, and build and support an ecosystem of complementary documentation, functionality, and services.

As a not-for-profit organization, OpenDDS Foundation relies on the financial support of contributing members to support and grow the project. Businesses and community members are encouraged to actively participate in the project's success by becoming contributing members through one of our sponsorship programs.

opendds.org/foundation

Q&A

THANK YOU!

LET'S CONNECT

☎ (314) 579-0066

✉ opendds@objectcomputing.com

🌐 opendds.org