UNIVERSITÉ DU LUXEMBOURG

ReFrame Regression Tests for Measuring Intra/Inter-Node Latency & Bandwidth

Presented by:
Franco Kušek, Heriel Harry SHAO, Ludovic TEMGOUA ABANDA N.

Contents

- Project Objectives
- Test Suite Creation & Execution Strategy
- Methodology & Implementation
- System & Topology Design
- Results (Latency & Bandwidth)
- Result Analysis & Key Takeaways
- Q&A

Project Objectives

Project Objectives

- **Objective 1**: Create regression tests for MPI communication using ReFrame and OSU Micro-Benchmarks.
- **Objective 2**: Capture effects of system architecture using hwloc and NUMA-aware bindings.
- **Objective 3**: Implement tests using multiple binary sources: source, EasyBuild, and EESSI.
- **Objective 4**: Validate on multiple clusters: Aion and Iris.
- **Objective 5**: Extract meaningful performance baselines using ReFrame's performance functions.

Test Suite Creation & Execution Strategy

Test Suite Creation & Execution Strategy

- **Script Organization:**
  - **Separated into source/, easybuild/, and eessi/ directories.**
  - **Each contains tests for `osu_latency`, `osu_bw`, and build stages.**
- **Parameterized Tests:**
  - **Implemented using `@rfm.simple_test` and `@require_deps`.**
  - **Test variants:** `same_numa`, `diff_numa_same_socket`, `diff_socket_same_node`, `inter_node`, `default`.

Test Suite Creation & Execution Strategy

- **System Bindings:**
  - **SLURM options (`--cpu-bind`, `--distribution`) and OMPI_MCA used.**
  - **Specific placement ensured via topology-aware settings.**
- **Execution Example:**

```
reframe -C reframe/configs/configs.py -c reframe/source -r --performance-report
```

UNIVERSITÉ DU LUXEMBOURG

Methodology & Implementation

Methodology & Implementation

- Benchmark Tool: OSU Micro-Benchmarks v7.2 (MPI-based)
- Automation Framework: ReFrame
- Message Sizes:
  - osu_latency: 8192 bytes
  - osu_bw: 1 MB
- Implementation Highlights:
  - Separate build and run classes.
  - Used performance functions to extract latency/bandwidth.
  - Reference values added to detect regressions.
- Output Parsing:
  - ReFrame compares results against reference thresholds.
  - Visual results available via `--performance-report`.

UNIVERSITÉ DU LUXEMBOURG

System Topology & Test Variants

System Topology & Test Variants

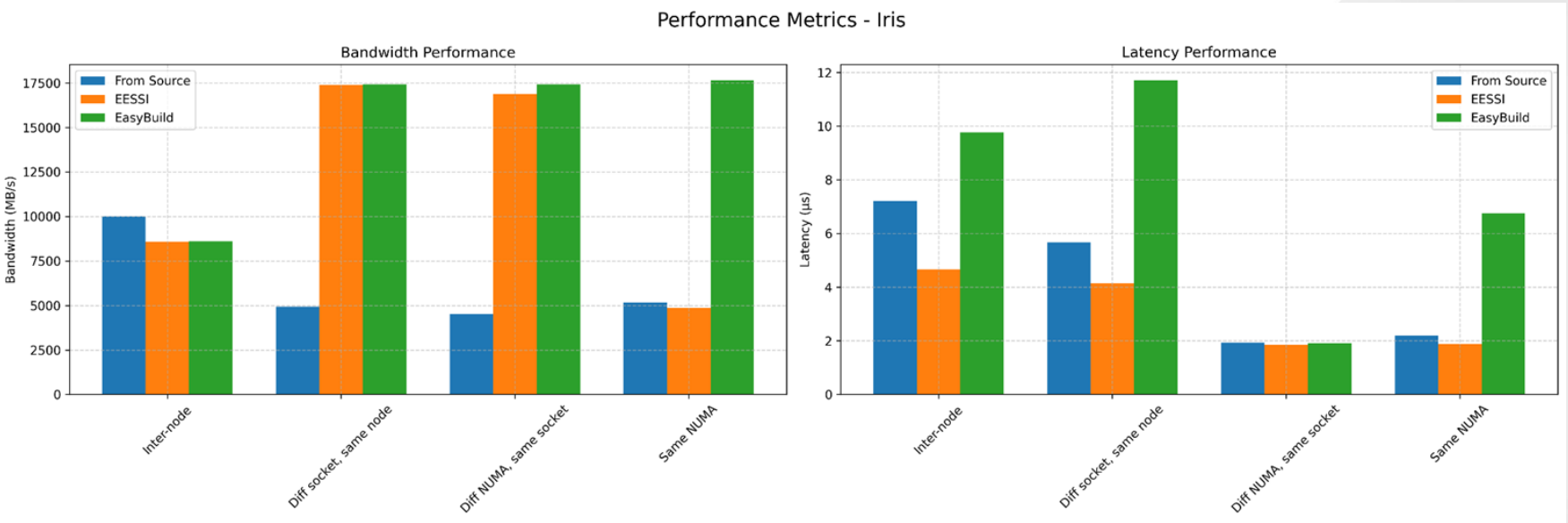
- Hardware Topology:
  - Explored using `lstopo` and `hwloc`.
- Variants Designed:
  - Same_numa
  - Diff_numa_same_socket
  - Diff_socket_same_node
  - Inter_node
- Binding Strategies:
  - Ensured proper CPU placement for each variant.
  - Controlled memory access paths to assess true communication overheads.

UNIVERSITÉ DU LUXEMBOURG

Results - Performance Metrics

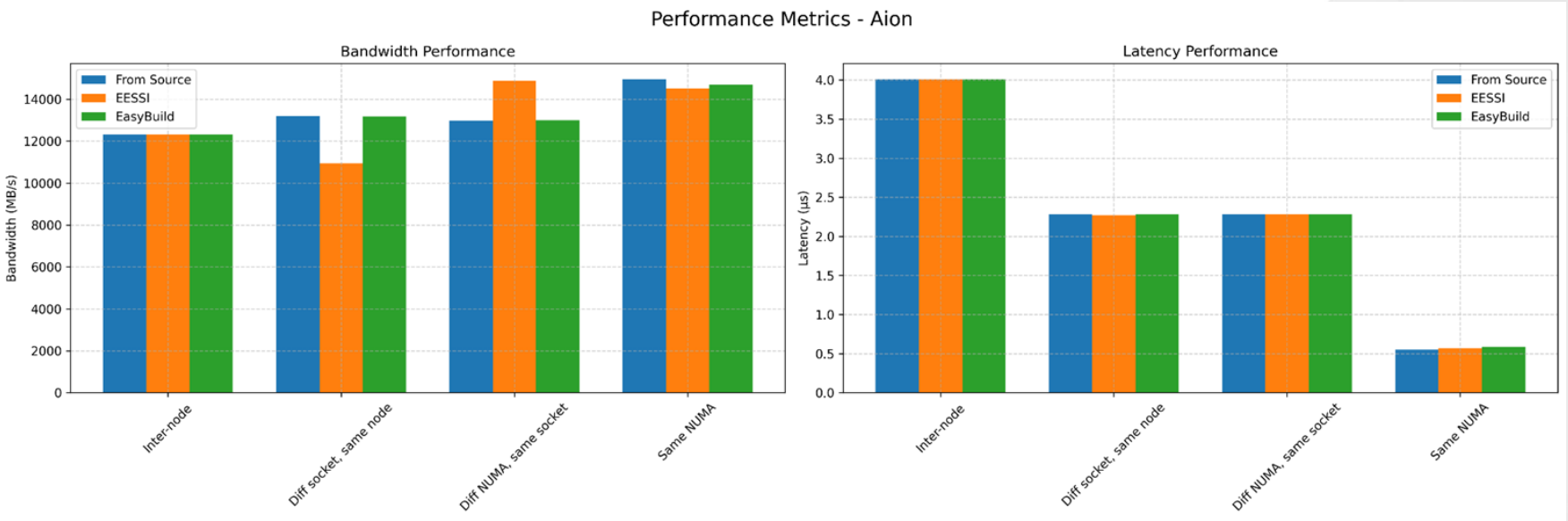
Results – Latency & Bandwidth (Iris)

Visual comparison of bandwidth (MB/s) and latency (µs) on Iris cluster



Results – Latency & Bandwidth (Aion)

Visual comparison of bandwidth (MB/s) and latency (µs) on Aion cluster



Analysis & Key Takeaways

Result Analysis & Key Takeaways

- Latency Insights:
  - Aion shows consistent sub-microsecond latency for local memory access.
  - Iris EasyBuild latency can spike in some inter-node cases.
- Bandwidth Observations:
  - EasyBuild delivers highest values on Iris (up to ~17,500 MB/s).
  - Aion maintains stable bandwidth ~14,500 MB/s on NUMA-local cases.
- Key Takeaways:
  - Binary source can affect consistency (e.g., EasyBuild vs EESSI).
  - Topology awareness is crucial for evaluating MPI performance.
  - ReFrame enables modular, repeatable regression test design.

UNIVERSITÉ DU LUXEMBOURG

Repository

- https://github.com/icemc/HPC-Environment-Project

Thank you for your attention!
We’re happy to answer your questions.