DArchR DAVIS ARCHITECTURE RESEARCH

Analyzing Google Workload Traces in gem5

Ayaz Akram, Maryam Babaie, Jason Lowe-Power
Computer Science Department, University of California Davis

Motivation

Warehouse scale computing (WSC) workloads

Google traces released recently for workloads like WebSearch, Ads, Fleet-Wide, Knowledge Graph [1,2]

WSC workloads have special characteristics different from traditional workloads

Simulating these traces can provide quick DSE for WSC architectures

Goal: Enable simulation of Google Workload Traces in gem5 and explore their behavior.

```
Want to try?

Visit:
https://github.com/darchr/gem5/tree/gtraces-gem5
```

Google Trace Player in gem5

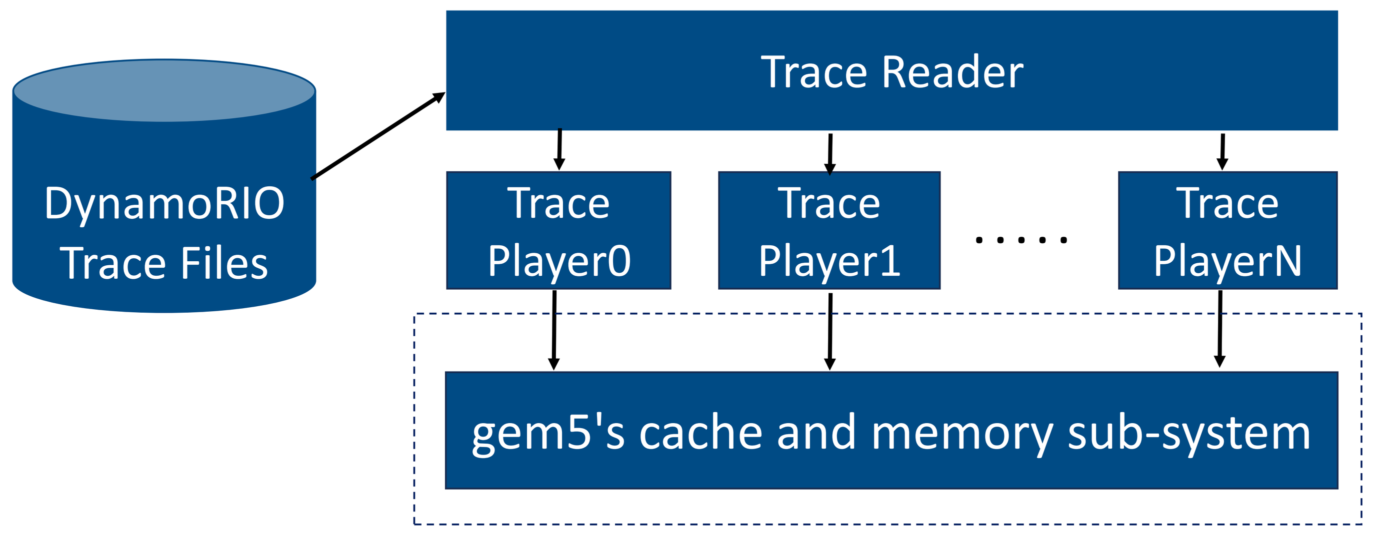
Components involved:

Trace Reader
- Traces in **drmemtrace** format
- Separate files for each software thread

Trace Player
- Configurable **max_ipc** and **max_outstanding_mem** reqs.

Trace Reader relies on timestamp information to pick a thread.
Trace feeds only **memory instructions** to trace players (configurable).

gem5 can simulate these traces at **1 million instructions per host second.**



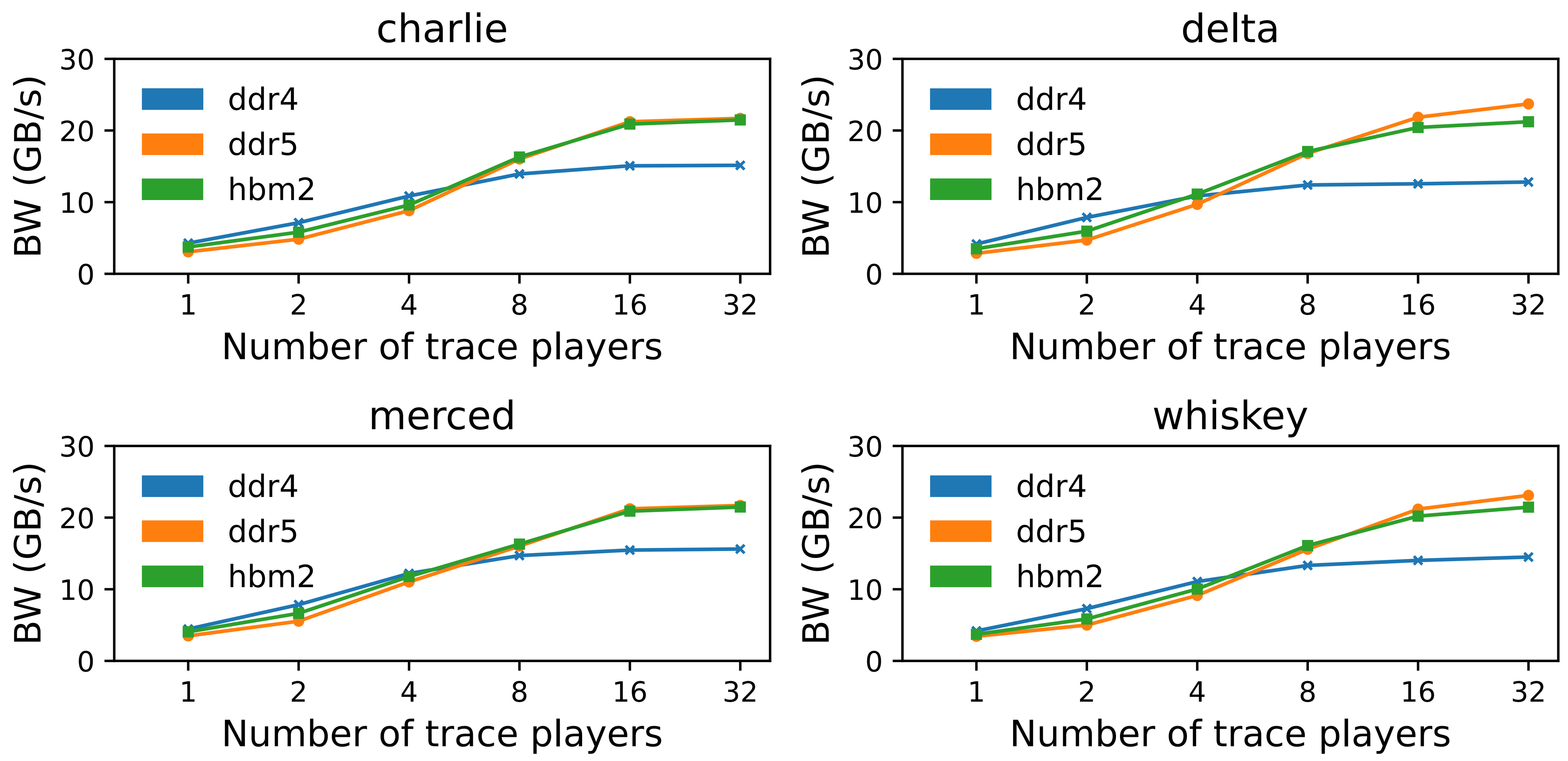
Overview of Google Trace Player in gem5

How much these traces stress memory systems?

How does the change in the number of trace players impact the observed bandwidth?

Configuration

Feature	Value
Cores	8
Core width	8
Frequency	5GHz
Private L1 I$	32KB
DRAM	HBM, DDR4/5



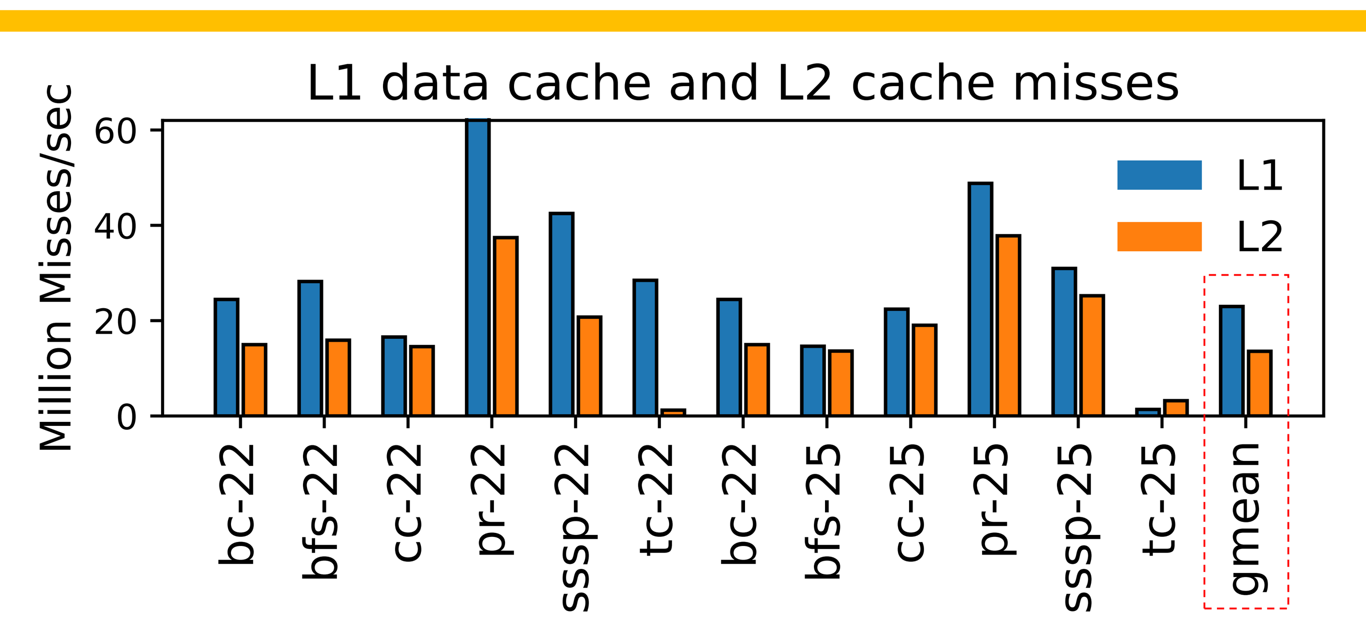
How do these traces compare with other HPC Benchmarks?

We compare cache miss rates of different benchmarks.

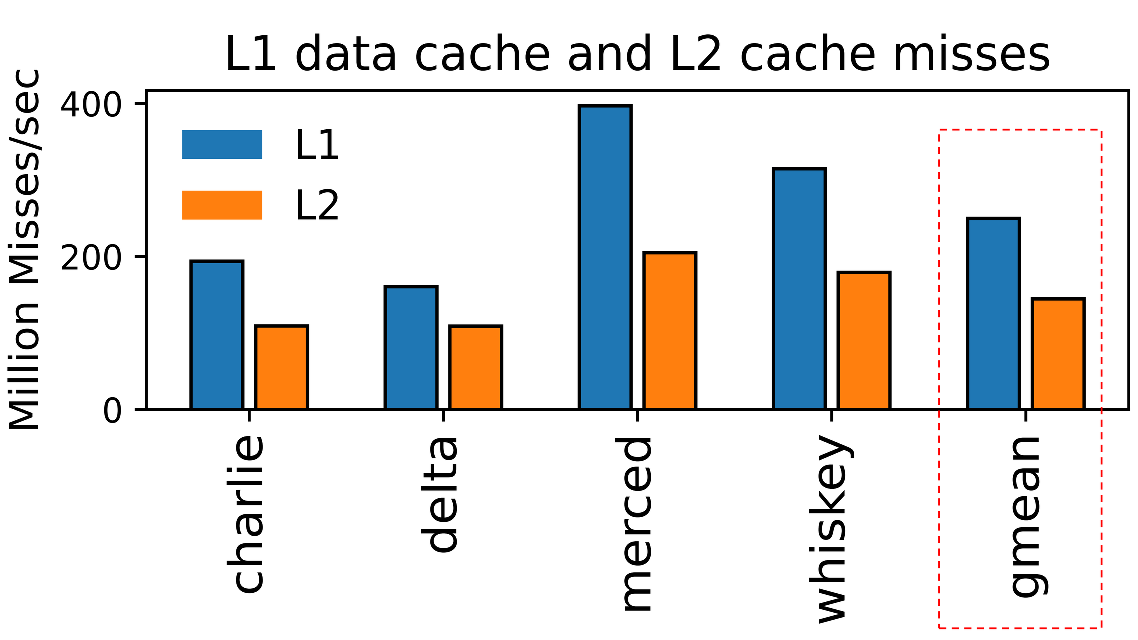
High cache miss rates indicate low locality in Google Workload Traces compared to traditional HPC benchmarks.

Configuration

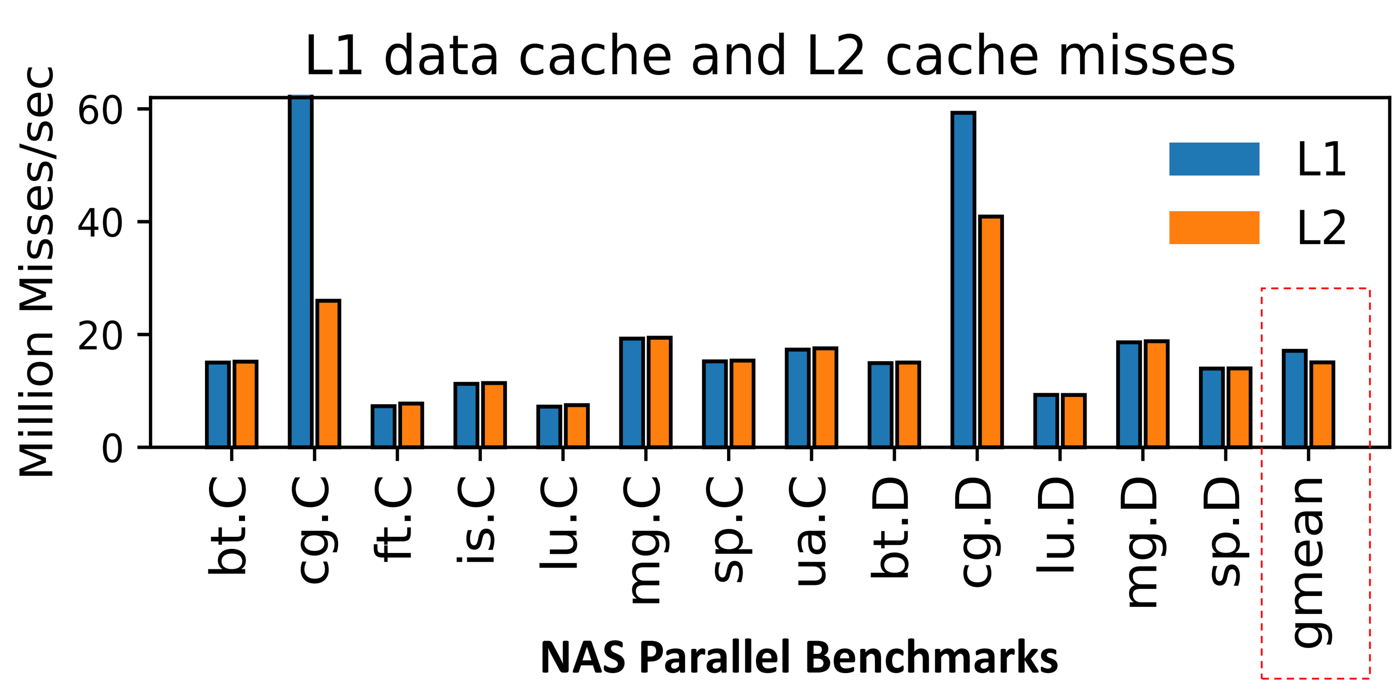
Feature	Value
Cores	8
Core width	8
Frequency	5GHz
Private L1 I$	32KB
Private L1 D$	512KB
Shared L2	8MB



GAP Benchmark Suite



Google Workload Traces



NAS Parallel Benchmarks

References

[1] https://dynamorio.org/google_workload_traces.html.
[2] Ayers et al., "Asmdb: understanding and mitigating front-end stalls in warehouse-scale computers" in ISCA 2019.