Scaling Parallelism under CPU-intensive Loads in Node.js

Maria Patrou, Kenneth B. Kent
University of New Brunswick, Faculty of Computer Science
Michael Dawson
IBM Canada
{maria.patrou, ken}@unb.ca, michael_dawson@ca.ibm.com

Node.js

- Server-side JavaScript environment on top of Google's V8 JavaScript Engine
- Asynchronous I/O
- Event-driven model – Single-threaded event loop

❖ Compute-intensive tasks depend on the performance of a **single** Core

Parallelization and Scaling Modules

- **Multi-Process**
  - Child Process
  - Cluster
- **Multi-Thread**
  - Napa.js
  - WebWorker-Threads

Different techniques produce different performance!

Motivation

Performance Efficiency. We need to determine which method is more appropriate for each case under scalable conditions.

Contribution

- Formulate a methodology
- Extract patterns
- Analyze and identify (dis)-similarities in **computational** performance
- Find the optimal techniques

Methodology

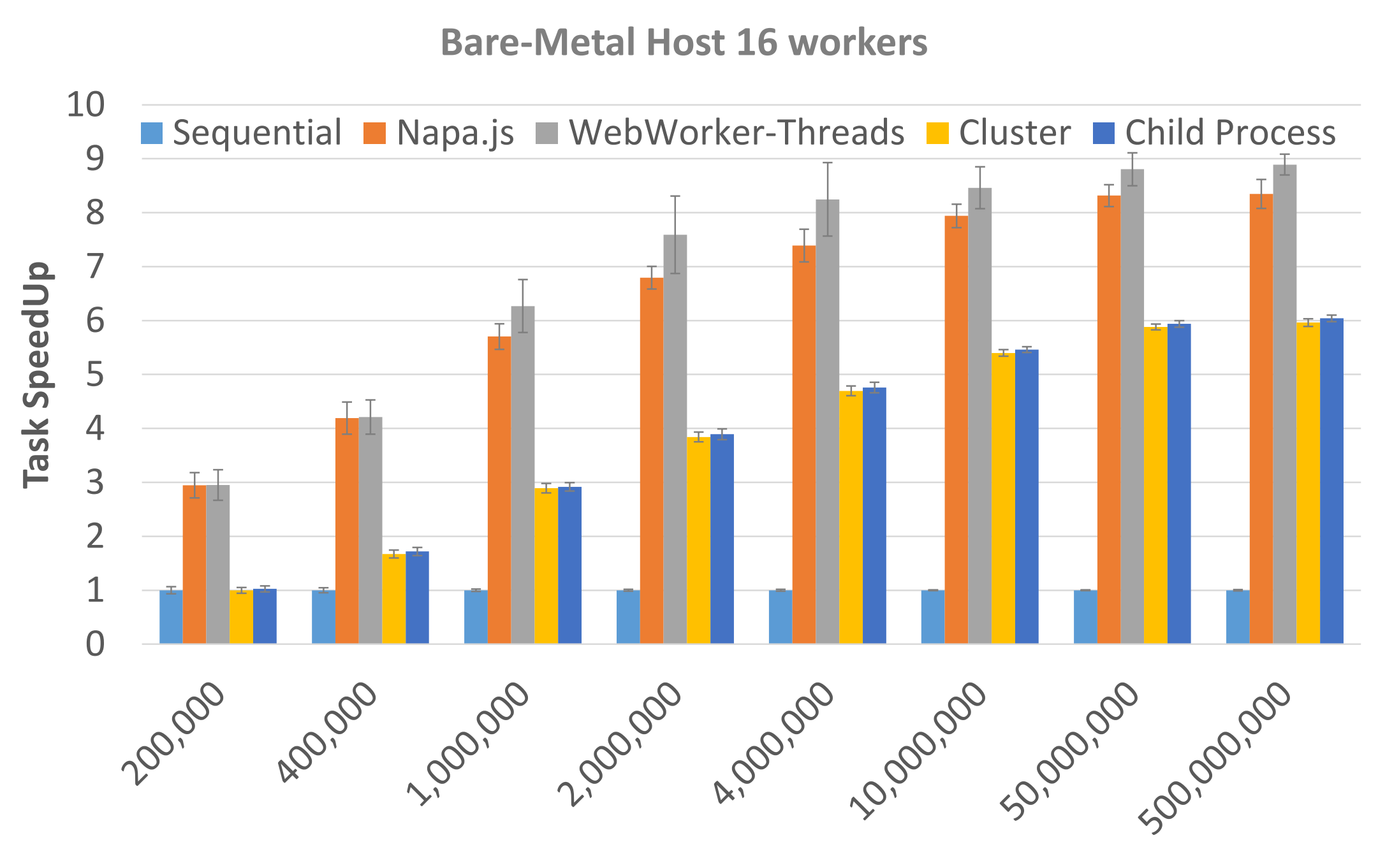
We use a compute-intensive task and vary:

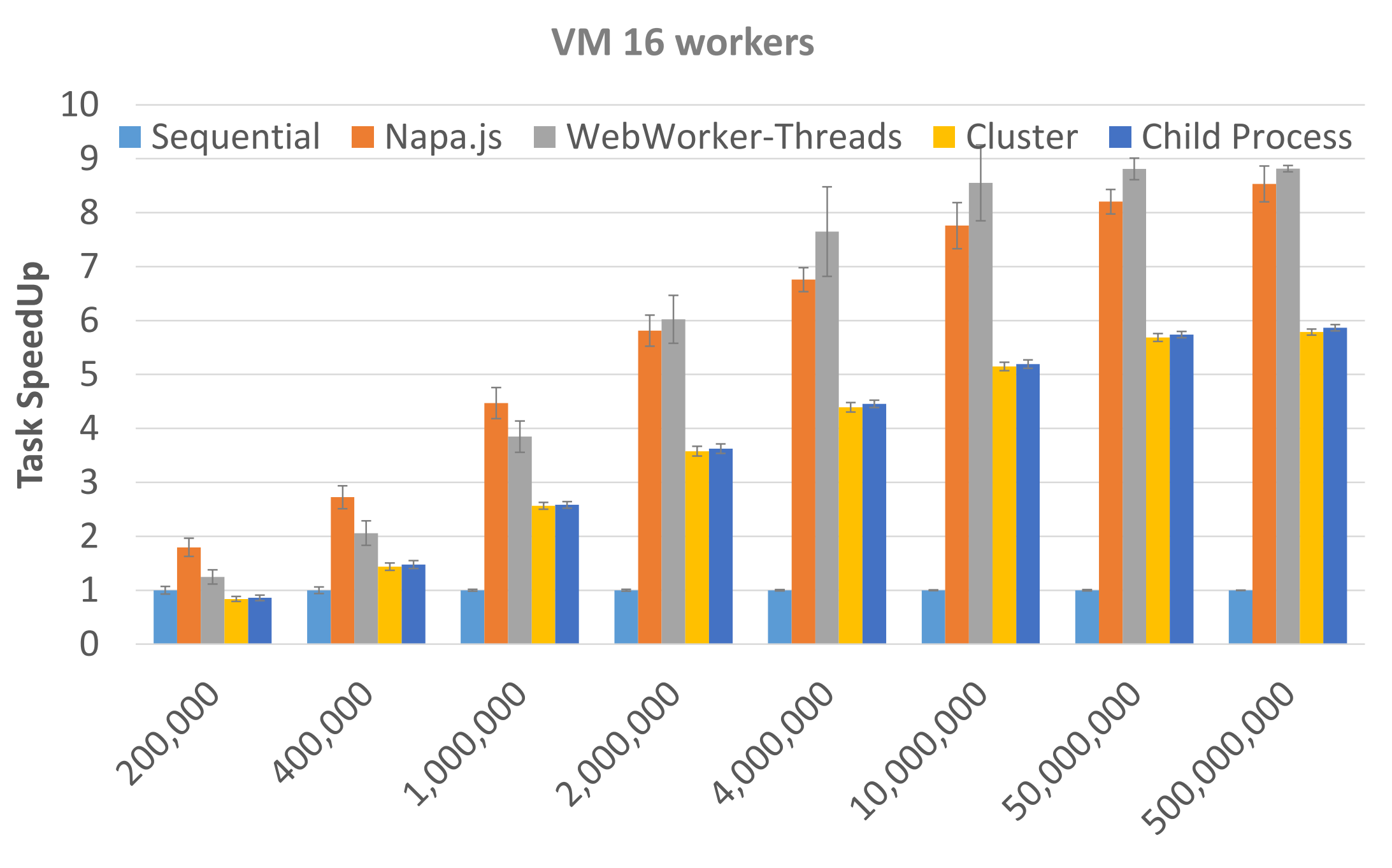
- *Task size in two dimensions;* number of instances and workload per instance
- *Execution environment*; bare-metal host vs. virtual environment.

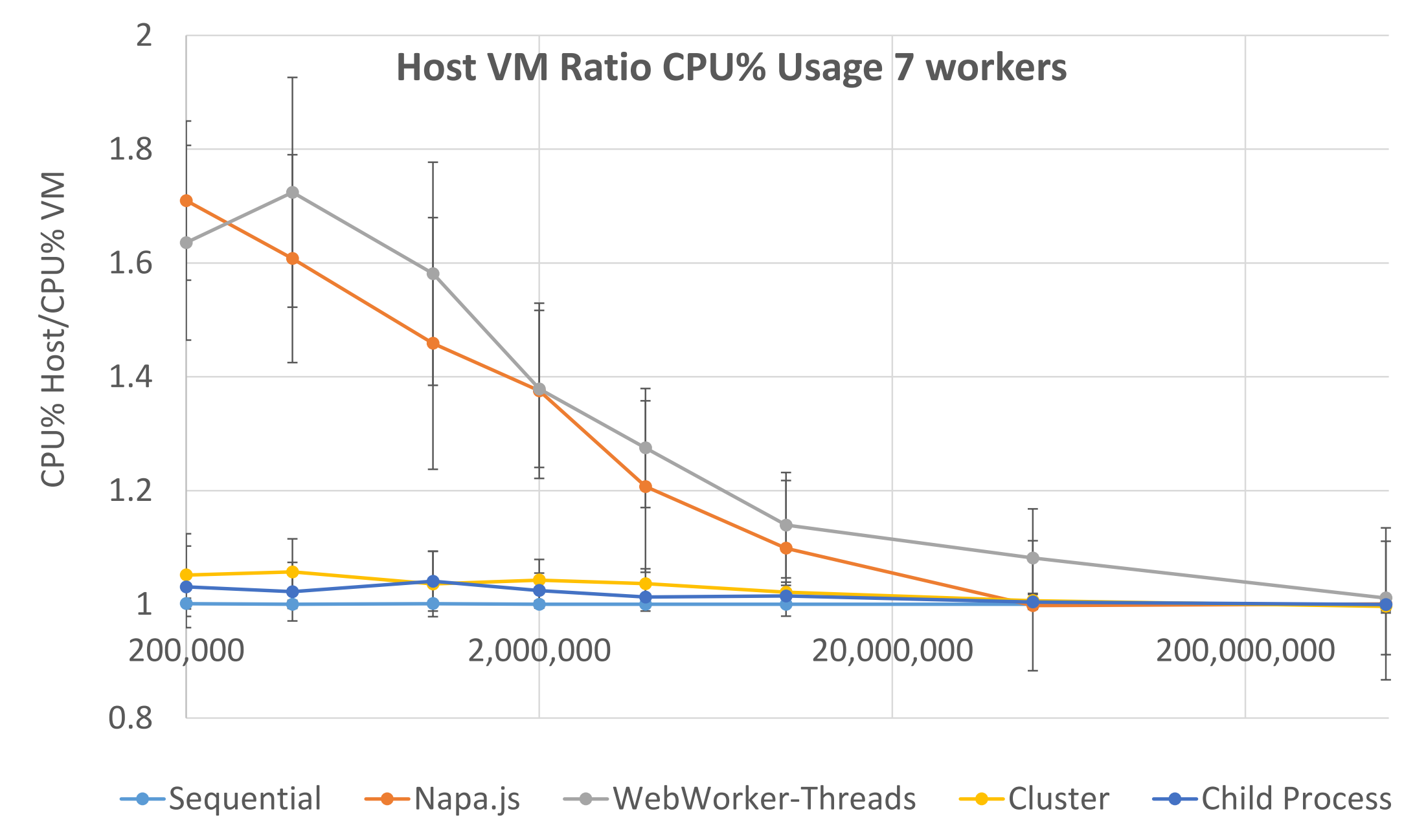
We collect data and present **performance metrics** with end goal to provide observations and recommendations.

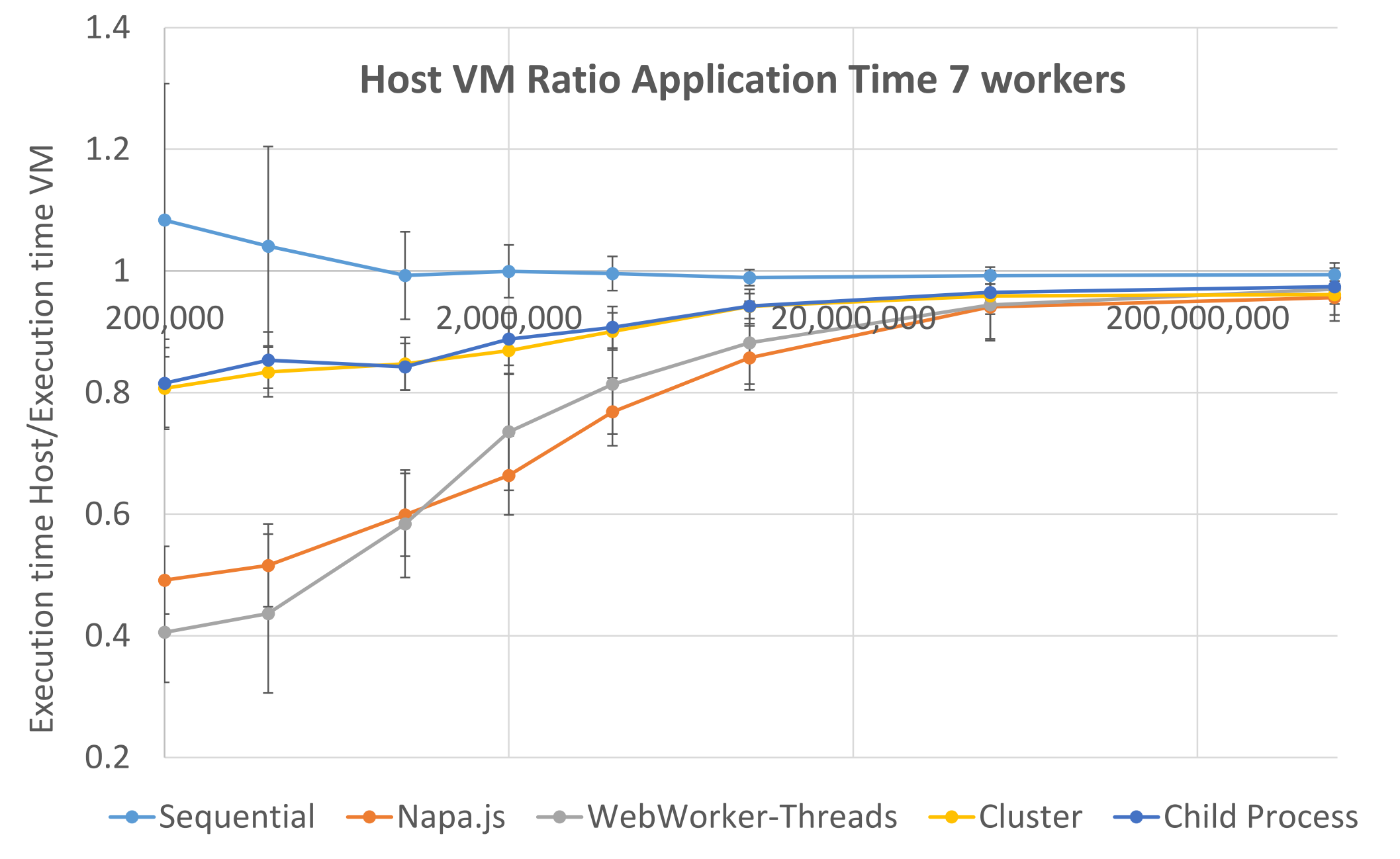
Performance Evaluation

❖ *SpeedUp – Higher is better*









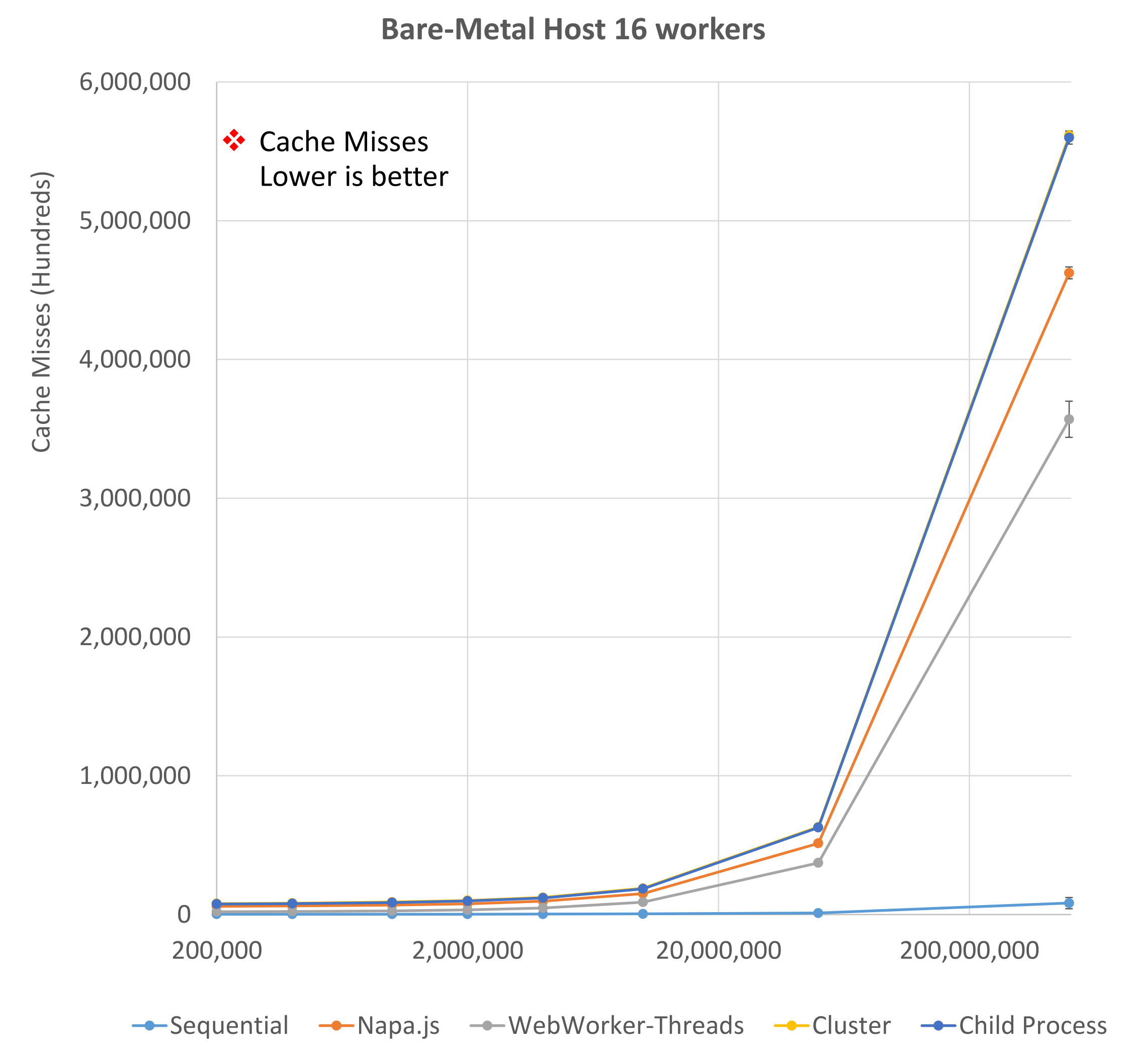
Observations & Recommendations

- Module implementation overcomes the instance type occasionally.

BUT at least one multi-thread technique produces better results than the multi-process ones.

- Multi-thread modules are more susceptible to environment for short-term applications

BUT the underlying environment does not change the overall trends



Conclusions & Future Work

- For a CPU-intensive task it is better to use a multi-thread approach considering the **computational** performance.

On-going research/methodology expansion:

- Heap usage, garbage collection patterns for every case
- Communication cost

UNB
EST. 1785
IBM Centre for Advanced Studies - Atlantic
FACULTY OF COMPUTER SCIENCE