I/O simulation extension for SimGrid framework

Hoang-Dung Do

April 15, 2020

Abstract

1 Introduction

2 Related Work

2.1 Page cache

- What is page cache? How it works.
- Effects and importance of page cache

2.2 Page cache replacement policies

- Introduce some proposed strategies.
- Some pros and cons of each algorithm if possible.

2.3 Implementation in Linux

- Details of the implemented strategy in Linux.
- Reason for this to be implemented. (implementation complexity, effectiveness, overhead, etc)

2.4 Simulators

Compare pros and cons of some simulators: SimGrid, GridSim (accuracy, simulation time, usability).

2.5 Our conclusion

3 Method

3.1 Principle of the simulator

- Objective: Add capability to simulate memory read/write, the impact of page cache on I/O in SimGrid.
- Approach: generalize dirty data, dirty ratio, cache eviction strategy implemented in Linux.

3.2 Implementation

- Which features of memory are implemented.
- Level of granularity, how features are implemented.
- Specific implementation detail in python and SimGrid.

3.3 Experiments

Describe data, workflow, number of tasks, task details, environment of each experiment.

3.3.1 Expriment 1

A single pipeline running one node.

3.3.2 Expriment 2

Multiple pipelines running in parallel on multiple nodes.

3.3.3 Expriment 3

Same as Experiment 2 but nodes write to a shared file system.

3.3.4 Expriment 4

A real pipeline (for example a pipeline with nighres)

4 Results

- Quantized results:
  - Errors of simulation time and memory used compared to real results.

  - Simulation time compared to baseline SimGrid.

- Ability of the model to generalize trends seen in memory (amount of dirty data, page cache) and disk throughput.

5 Discussion