

Torpor: Controlled Environments for The Reproducible Evaluation of Systems

Abstract

The key contributions of a paper in systems research is the insight, not the absolute performance numbers. Readers value the high-level goals and findings of a paper, and intuitively interpret experiments. Torpor is a framework for creating controlled experiments where insights can be reproduced. It is based on a simple idea: for every resource used of the underlying computing platform where an experiment is executed (MEM/CPU/IO/NET), take the lowest common denominator. For intraplatform reproducibility this means taking the slowest node and reproducing its performance in other faster nodes. For cross-platform reproducibility, it means taking the slowest cluster and calibrating other platforms with respect to it.

1. Intro

Reproducing insights within and across platforms is hard. Torpor helps.

2. Meat

Our approach:

- Run microbenchmarks that characterize the performance of the underlying resources: CPU (stress-ng), IO (fio) and network (conceptual). We call this the performance profile (PP) of the platform.
- Obtain PP B and T for base and target platforms, respectively. Base being the platform where an experiment was originally executed and target where it will get re-executed.
- Based on PPs PP_b and PP_t we calibrate T using Torpor and obtain a set of throttling parameters for each resource (CPU, IO and network) of T . Since we assume that T is

more capable than B for all available resources, we can always find parameters that will slow down T ¹.

Caveats: What about ratios? E.g. 4:1:2 (4x faster on CPU, same IO performance and 2x faster on network)? We assume that throttling CPU doesn't affect IO/net, but it might. Also, what happens when the slowest common denominator is not constant?

- We re-run on platform T and observe results that validate the experiment originally executed on B .

3. Experiments

3.1 Intraplatform Variability

- Ceph (IO-/network-bound). **platform:** issdm.
- CPU-bound workload. **platform:** amazon.

3.2 Cross-platform Reproducibility

- MPI (CPU/network intensive)
- GassyFS (MEM/network intensive)

Platforms:

- cloudlab w.r.t. issdm
- chameleon w.r.t. issdm
- amazon w.r.t. issdm

Bibliography

¹ We named our framework Torpor as a reference to the state of decreased physiological activity that some animals undergo in order to reduce energy consumption. In our case, we reduce activity in order to slowdown performance with the goal of minimizing variability within and across platforms