Motivating Work

Structure is revealed in the curve of varying energy/performance points achieved as the system responds to controlled changes of ITR and DVFS settings.

We deduce that network-bound software stacks, influenced externally by varying QPS rates, will operate within different timescales, or time signatures, defined by an inter-response frequency and intra-response rate.

We believe that there exists a timescale for which a target execution consumes minimal energy and exhibits optimal or sub-optimal performance.

We propose impedance matching, or rate matching, as the missing system primitive that would match the internal timescale of the system, via energy/performance configuration, to the external timescale of the world it is responding to.

Idea

We believe that network-bound software stacks, influenced externally by varying QPS rates, will operate within different timescales, or time signatures, defined by an inter-response frequency and intra-response rate.

We deduce that there exists a timescale for which a target execution consumes minimal energy and exhibits optimal or sub-optimal performance.

We propose impedance matching, or rate matching, as the missing system primitive that would match the internal timescale of the system, via energy/performance configuration, to the external timescale of the world it is responding to.

Approach

The methodology lies in developing a system component that is able to learn the properties of ideal timescales for a set of software stacks and 2. automatically improve the system’s energy and performance by configuring it to that ideal timescale under any QPS rate.

This component must be designed as a dynamic control mechanism, based on a model that can learn from a time-varying, multidimensional, signal-based interpretation of the target execution.

Objectives: Toward an Architecture for Learnable Energy/Performance Control Policies

1. Encoding

From an execution log to a multi-dimensional vector representation of the log’s energy and performance characteristics.

1. Manual statistics, characterizing logs through their constituent percentage.
2. Automated statistics, computed by a recurrent neural network (RNN) as it observes execution logs across time and produces a vector representative of the full execution.

2. Learning

From the state-space of execution encoding to a mapping between execution and optimal ITR/DVFS settings.

A Bayesian optimization algorithm traversing different curves of energy/performance targets, where each curve corresponds to the software stack behavior subject to a different QPS rate.

A curve of energy/performance achieved. When directed to do so, a Bayesian optimization algorithm is able to exploit this relationship to find the most optimal settings for execution.

3. System Regulation

From one execution state to another, improving energy performance through a more optimal setting of ITR and DVFS.

For an arbitrary (app, os, cpu, nic), the controller has knowledge about optimal execution settings for arbitrary QPS rates such that it is able to always set the system settings for an optimal energy/performance.