Supplementary Data

Table 1: Configurable software systems studied.

Software	Domain	Performance Objective	$\lvert\mathcal{O}\rvert$	Search Space
Trimesh	Mesh	O1: # Iteration; O2: Latency	13	239,260
x264	Video	O1: PSNR; O2: Energy Usage	17	53,662
Storm/WC	SP	O1: Throughput; O2: Latency	6	2,880
Storm/RS	SP	O1: Throughput; O2: Latency	6	3,839
Storm/SOL	SP	O1: Throughput; O2: Latency	13	2,048
Keras-DNN/DSR	DL	O1: AUC; O2: Inference Time	13	3.32×10^{13}
Keras-DNN/Coffee	DL	O1: AUC; O2: Inference Time	13	2.66×10^{13}
Keras-LSTM	DL	O1: RMSE; O2: Inference Time	13	7,040

Extending the information from Table 1, the real-world software systems and environments considered in the work are as follow:

- Trimesh: This is a Python library for loading and using triangular meshes with an emphasis on watertight surfaces. The goal is to provide a full featured and well tested Trimesh object which allows for easy manipulation and analysis, in the style of the Polygon object in the Shapely library. We use the standard benchmark that consists of a Trimesh object with a moderate degree of complexity.
  - **Configuration Options:** We consider options such as F, smoother, colorGS, relaxParameter, V, Jac- obi, line, and zebraLine etc.
  - **Performance Objectives:** We consider two performance objectives: the latency of completing a mesh (to minimize) and the number of iteration that represent the fidelity of a mesh (to maximize).
- x264: A free and open-source software library and a command-line utility developed by VideoLAN for encoding video streams into the H.264/MPEG-4 AVC video coding format We use a randomly chosen video that is around 1GB as the benchmark.
  - **Configuration Options:** We consider options such as no_mbtree, no_asm, no_cabac, and no_scenecut, etc.
  - **Performance Objectives:** We consider two performance objectives: the energy usage (which is proportionally related to the CPU

usage) of a compression (to minimize) and the PNSR— ratio between the maximum possible power of a signal and the power of corrupting noise— to represent the quality of compression (to maximize).

- STORM/WC: A distributed stream processing computation framework, using custom created "spouts" and "bolts" to define information sources and manipulations to allow batch, distributed processing of streaming data. We use the standard benchmark named WORDCOUNT, which is CPU intensive.
  - **Configuration Options:** We consider options spouts, max_spout, spout_wait, no_spliters, no_counters, and netty_min_wait.
  - **Performance Objectives:** We consider two performance objectives: the average latency of all the messages processed (to minimize) and the throughput that is about the number of messages processed over a period of time (to maximize).
- STORM/RS: The same as STORM/WC, but we use the standard benchmark named ROLLINGSORT, which is memory intensive.
  - **Configuration Options:** We consider options spouts, max_spout, sorters, emitfreq, chunksize, and message size.
  - **Performance Objectives:** The same as STORM/WC.
- STORM/SOL: The same as STORM/WC, but we use the standard benchmark named SOL, which is network intensive.
  - **Configuration Options:** We consider options such as topology.workers, component.bolt_num, topology.acker.executors, component.spout_num, topology.transfer.buffer.size, and topology.priorit, etc.
  - **Performance Objectives:** The same as STORM/WC.
- KERAS-DNN/DSR: Keras is an open-source software library that provides a Python interface for artificial neural networks. It often acts as an interface for the TensorFlow library. In this work, we apply Deep Neural Network (DNN) as the actual deep learning model. We use the a set of random samples from the DiatomSizeReduction dataset as the benchmark environment.
  - **Configuration Options:** We consider options such as log_beta_2, log_decay, batch_size, conv_1_num_filters, conv_1_filter_size, and conv_2_num_filters, etc.
  - **Performance Objectives:** We consider two performance objectives: the AUC which is a robust metrics to measure the quality of a classifier (to maximize) and the inference time which denotes the time taken to make a prediction (to minimize).

- Keras-DNN/Coffee: The same as Keras-DNN/DSR, but we use the a set of random samples from the Coffee dataset as the benchmark environment.
  - **Configuration Options:** We consider options such as log_beta_1, log_decay, batch_size, conv_1_num_filters, and conv_2_num_filters, etc.
  - **Performance Objectives:** The same as Keras-DNN/DSR.
- Keras-LSTM: The same as Keras-DNN/DSR, but the deep learning model is Long-Short Term Memory (LSTM) network. We use the a set of random samples from the Trans-Atlantic Slave Trade dataset as the benchmark environment.
  - **Configuration Options:** We consider options such as size, num_neurons, num_neurons_layer_1, num_neurons_layer_2, and dropout, etc.
  - **Performance Objectives:** We consider two performance objectives: the RMSE which is a robust metrics to measure the quality of a regressor (to minimize) and the inference time which denotes the time taken to make a prediction (to minimize).