Data-Calculator-Examples

Data Systems Lab, Harvard University

April 2020

1 Example 1: Arrays

In this example, we first demonstrate how to use Data Calculator for costing an array data structure. We specify the inputs to be fed to the calculator and the outputs that we obtain from it.

1.1 Input to the Calculator

The input to the calculator is an array of data layout primitives in the format of `<primitive>,<value>`. For each primitive, we set a value chosen from the domain of the primitive. A complete input specification of an array variant is shown below.

Listing 1: Data Structure Specification for Arrays


```
"external.links.next":False,
"external.links.prev":False,
"inter-block.blockAccess.direct":True,
"inter-block.blockAccess.headLink":False,
"inter-block.blockAccess.tailLink":False,
"inter-block.fanout.fixedValue":1,
"inter-block.fanout.function":"",
"inter-block.fanout.type":"fixed",
"inter-block.orderMaintenance.type":"push&insert",
"inter-block.partitioning.function":"",
"inter-block.partitioning.logStructured. filtersPerLevel ":False,
"inter-block.partitioning.logStructured.filtersPerRun":False,
"inter-block.partitioning.logStructured.initialRunSize":0,
"inter-block.partitioning.logStructured.maxRunsPerLevel":0,
"inter-block.partitioning.logStructured.mergeFactor":0,
"inter-block.partitioning.type":"none",
"intra-block.blockProperties.storage":"none",
"intra-block.capacity.function":"",
"intra-block.capacity.type":"balanced",
"intra-block.capacity.value":10,
"intra-block.dataRetention.keyRetention.compression":"uncompressed",
"intra-block.dataRetention.keyRetention.function":"",
"intra-block.dataRetention.keyRetention.type":"full",
"intra-block.dataRetention.retainedDataLayout":"row-wise",
"intra-block.dataRetention.valueRetention.compression":"uncompressed",
"intra-block.dataRetention.valueRetention.function":"",
```

```
"intra-block.dataRetention.valueRetention.type":"full",
"intra-block.filters.bloomFilter.active":False,
"intra-block.filters.bloomFilter.hashFunctionsNumber":0,
"intra-block.filters.bloomFilter.numberOfBits":0,
"intra-block.filters.filtersMemoryLayout":"scatter",
"intra-block.filters.zoneMaps.max":False,
"intra-block.filters.zoneMaps.min":False,
"intra-block.links.linksMemoryLayout":"scatter",
"intra-block.links.next":False,
"intra-block.links.prev":False,
"intra-block.links.skipLinks.probability":0,
"intra-block.links.skipLinks.type":"none",
"intra-block.utilization.constraint":"leq_capacity",
"intra-block.utilization.function":"",
"metadata.general.name":"ARRAY"
```


You may either pass this a command line input to `data-calculator/cost-synthesis/homomorphic_coster.py` or you may set the desired definition within the variable `sample_node_defns` within `data-calculator/cost-synthesis/homomorphic_coster.py`.

1.2 Cost Synthesis within the Calculator

For synthesizing cost for running a workload with an input data structure specification, there are learned data that are pre-generated at the time of installation of the calculator on a given hardware. At that time, the calculator has learned the hardware-aware coefficients of the cost models to be used later during cost synthesis. An example representation of the learn coefficients is shown below.

1.3 Workload and Hardware

For this example, we are using a workload comprising of 10000 puts and 10000 gets. We use a 2.3 GHz Quad-Core Intel Core i5 MacBook Pro with 8 GB 2133 MHz LPDDR3 memory.

1.4 Output From the Calculator

The output obtained from the calculator is a single value representing predicted latency to run the workload (in second). The cost of running the aforementioned workload on the specified hardware is $4.1185e-07$ seconds.

2 Example 2: Linked Lists

We further demonstrate the next example with linked list variants. We use the same workload and hardware for this example. The input to the calculator is specified below.

Benchmarks	Models	Coefficients (space separated)
SimdAVXScanEqualityContiguousKeyBenchmark	SimpleLinearModel	1.87187e-10 1.25388e-08
SimdAVXScanRangeContiguousKeyBenchmark	SimpleLinearModel	7.52889e-10 0
ExternalMergeSortBenchmark	NLogNModel	1.23036e-08 -6.15347e-08 -8.30936e-06
MergeSortBenchmark	NLogNModel	9.41085e-09 6.98588e-08 9.71407e-05
QuickSortBenchmark	NLogNModel	4.3305e-09 7.02713e-10 -7.29134e-06
NoSimdScanEqualityContiguousKeyValueBenchmark	SimpleLinearModel	1.6525e-10 -5.45448e-09
NoSimdScanRangeContiguousKeyValueBenchmark	SimpleLinearModel	2.34468e-09 0
NoSimdScanRangeContiguousKeysBenchmark	SimpleLinearModel	2.41797e-09 0
BinarySearchContiguousKeysBenchmark	LinearPlusLogModel	1.93111e-13 9.96276e-09 -1.70091e-08
BinarySearchContiguousKeysBenchmarkLarge	LinearPlusLogModel	4.46912e-15 3.66231e-08 -3.29354e-07

Table 1: Hardware-Aware Learned Coefficients for Cost Models

Listing 2: Data Structure Specification for Linked Lists

```
"external.links.next":False,
"external.links.prev":False,
"inter-block.blockAccess.direct":False,
"inter-block.blockAccess.headLink":True,
"inter-block.blockAccess.tailLink":False,
"inter-block.fanout.fixedValue":1,
"inter-block.fanout.function":"",
"inter-block.fanout.type":"fixed",
"inter-block.orderMaintenance.type":"push&insert",
"inter-block.partitioning.function":"",
"inter-block.partitioning.logStructured. filtersPerLevel ":False,
"inter-block.partitioning.logStructured.filtersPerRun":False,
"inter-block.partitioning.logStructured.initialRunSize":0,
"inter-block.partitioning.logStructured.maxRunsPerLevel":0,
"inter-block.partitioning.logStructured.mergeFactor":0,
"inter-block.partitioning.type":"none",
"intra-block.blockProperties.storage":"none",
"intra-block.capacity.function":"",
"intra-block.capacity.type":"balanced",
"intra-block.capacity.value":10,
"intra-block.dataRetention.keyRetention.compression":"uncompressed",
"intra-block.dataRetention.keyRetention.function":"",
"intra-block.dataRetention.keyRetention.type":"full",
"intra-block.dataRetention.retainedDataLayout":"row_wise",
```

"intra−block.dataRetention.valueRetention.compression":"uncompressed",
"intra−block.dataRetention.valueRetention.function":"",
"intra−block.dataRetention.valueRetention.type":"full",
"intra−block. filters .bloomFilter.active":False,
"intra−block. filters .bloomFilter.hashFunctionsNumber":0,
"intra−block. filters .bloomFilter.numberOfBits":0,
"intra−block. filters .filtersMemoryLayout":"scatter",
"intra−block. filters .zoneMaps.max":False,
"intra−block. filters .zoneMaps.min":False,
"intra−block.links.linksMemoryLayout":"scatter",
"intra−block.links.next":True,
"intra−block.links.prev":False,
"intra−block.links.skipLinks. probability":0,
"intra−block.links.skipLinks.type":"none",
"intra−block. utilization .constraint":"leq_capacity",
"intra−block. utilization .function":"",
"metadata.general.name":"LINKED LIST"

The output from the calculator is obtained as $2.05925e-07$ seconds.