Report Week 9

Achievements:

Parallelism achieved:

```
|0%                          50%                         100%|
 ============================================================
2017-11-21 18:32:03 INFO: Time 0:00:00.348487 taken by RouterProvenanceGatherer
Getting profile data
|0%                          50%                         100%|
 ============================================================
2017-11-21 18:32:03 INFO: Time 0:00:00.004204 taken by ProfileDataGatherer
2017-11-21 18:32:03 INFO: 0, 0, 2 > 15
2017-11-21 18:32:03 INFO: 0, 0, 3 > 1
2017-11-21 18:32:03 INFO: 0, 0, 4 > 1
2017-11-21 18:32:03 INFO: 0, 0, 5 > 1
2017-11-21 18:32:03 INFO: 0, 0, 6 > 1
2017-11-21 18:32:03 INFO: 0, 0, 7 > 1
2017-11-21 18:32:03 INFO: 0, 0, 8 > 1
2017-11-21 18:32:03 INFO: 0, 0, 9 > 1
2017-11-21 18:32:03 INFO: 0, 0, 10 > 1
2017-11-21 18:32:03 INFO: 0, 0, 11 > 1
2017-11-21 18:32:03 INFO: 0, 0, 12 > 1
2017-11-21 18:32:03 INFO: 0, 0, 13 > 1
2017-11-21 18:32:03 INFO: 0, 0, 14 > 1
2017-11-21 18:32:03 INFO: 0, 0, 15 > 1
2017-11-21 18:32:03 INFO: 0, 0, 16 > 1
2017-11-21 18:32:03 INFO: 0, 0, 17 > 1
```

The leader (core 0,0,2) sends an integer with value 1 to all other cores at the same time, and all of those cores return the integer back to the leader simultaneously. Then, the leader takes all of those returned integers and adds them up before returning the result to the host.

Here, the communication structure looks as follows:

```
 ------------------------------------------- [02]-------------------------------------------
  |     |     |     |     |     |     |      |     |     |     |     |     |     |     |
[03] [04] [05] [06] [07] [08] [09] [10] [11] [12] [13] [14] [15] [16] [17]
```

Considering problems of scalability

The next logical steps in building the database would be:

1. implementing the previously discussed histogram functioning
2. connecting several chips

In theory, implementing point 1 and 2 is easily doable with the current results. Unfortunately, there are several problems with scalability; for instance unique identifiers for every entry within SDRAM can only take up to 64Kbytes of TCM, which puts a limit on the number of entries that can be managed by the cores using this method.

Different issues have to be addressed in the foreseeable future, such as an alternative to using integer Ids, string size management and many more, including the problem of instability that comes with using UDP connections.

	Memory per processor				
	Bits	Bytes	KB	MB	GB
SDRAM	8,589,934,592	8,388,608	8,192	8	0.00781
TCM	67,108,864	65,536	64	0.06250	0.00006
	Memory per chip (1 SDRAM, 16 processors)				
SDRAM	137,438,953,472	134,217,728	131,072	128	0.12500
TCM	1,073,741,824	1,048,576	1,024	1	0.00098
	Memory per board (48 chips)				
SDRAM	6,597,069,766,656	6,442,450,944	6,291,456	6,144	6
TCM	51,539,607,552	50,331,648	49,152	48	0.04688

String sizes and maximum amount of possible entries in this format				
Number of ASCII	Bits	Bytes	Max num strings per TCM	per SDRAM*
4	32	4	16,384	16,777,216
8	64	8	8,192	8,388,608
12	96	12	5,461	5,592,405
16	128	16	4,096	4,194,304
20	160	20	3,277	3,355,443
24	192	24	2,731	2,796,203
28	224	28	2,341	2,396,745
32	256	32	2,048	2,097,152

Distribution of string entries on SDRAM slices					
Number of ASCII	SDRAM/16	5 Columns	10 Columns	20 Columns	50 Columns
4	1,048,576	209,715	104,858	52,429	20,972
8	524,288	104,858	52,429	26,214	10,486
12	349,525	69,905	34,953	17,476	6,991
16	262,144	52,429	26,214	13,107	5,243
20	209,715	41,943	20,972	10,486	4,194
24	174,763	34,953	17,476	8,738	3,495
28	149,797	29,959	14,980	7,490	2,996
32	131,072	26,214	13,107	6,554	2,621

Possible number of integers in TCM			
Number of Columns	32 int per TCM	16 int per TCM	8 int per TCM
1	2,097,152	4,194,304	8,388,608
2	1,048,576	2,097,152	4,194,304
4	524,288	1,048,576	2,097,152
8	262,144	524,288	1,048,576
16	131,072	262,144	524,288
32	65,536	131,072	262,144
64	32,768	65,536	131,072
128	16,384	32,768	65,536

*Here, we use 50% of SDRAM for storing the data entries – part of SDRAM has to be used for other purposes, e.g. writing output, storing connection data, iobuf etc.