MPI NAS Parallel Benchmarking

CSC 569

Ian Dunn
Computer Science Department
Cal Poly
San Luis Obispo, California
idunn01@calpoly.edu

Toshi Kuboi
Computer Science Department
Cal Poly
San Luis Obispo, California
tkuboi@calpoly.edu

Mitchell Rosen
Computer Science Department
Cal Poly
San Luis Obispo, California
mwrosen@calpoly.edu

Austin Wylie
Computer Science Department
Cal Poly
San Luis Obispo, California
awylie@calpoly.edu

1. INTRODUCTION

With the goal of exploring the performance benefits and drawbacks of distributed computing, we set up and ran various computationally-distributable benchmarks on two cluster types. A set of the Numerical Aerodynamic Simulation (NAS) Parallel Benchmarks (NPB) were run on both the machines in lab 14-302 (with 1 to 32 nodes) and Raspberry Pis (with 1 to 5 nodes). The Message Passing Interface (MPI) library was responsible for inter-machine communication. As an extra note, the lab machines have 3 GHz processors and 4 GB of memory; the Rasperry Pis have 700 MHz processors and 512 MB of memory.

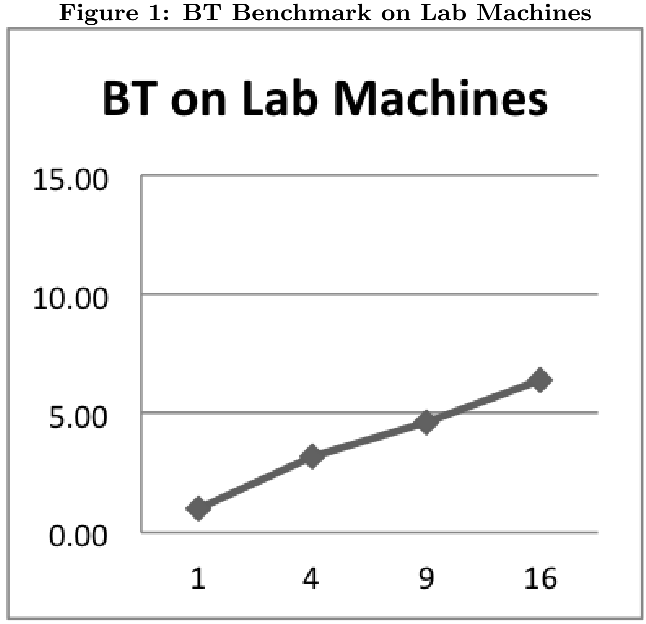
2. RESULTS

The primary metric we explore in the benchmarks is speedup, which is calculated by dividing the time taken by the benchmark on the fewest number of nodes (usually one or two) by the time taken by the current run. We could comment on the raw runtimes of these different scenarios and the effect of different processors and RAM setups, but speedup best quantifies the improvement from actually distributing the work.

Tables 1 and 2 shows the NPB results for the lab machine cluster and the Pi cluster, including the specific benchmark, runtime in seconds, speedup, millions of operations per second (MOPS), and MOPS per process.

Note that, in the graphs, vertical axes show the speedup and horizontal aces show the number of nodes the benchmark was run on.

Figure 1: BT Benchmark on Lab Machines



2.1 Lab Machines

Table 1 shows the lab machines' NPB results. The class B benchmarks were used.

Figures 1 through 6 graph those speedup results.

Table 1: Lab Machine Benchmark Results

Benchmark	Nodes	Time	Speedup	MOPS	Process MOPS
BT	1	339.12	1.00	2070.57	2070.57
BT	4	106.65	3.18	6583.81	1645.95
BT	9	73.82	4.59	9512.50	1056.94
BT	16	53.13	6.38	13215.32	825.96
FT	1	90.52	1.00	1016.91	1016.91
FT	2	76.86	1.18	1197.60	598.80
FT	4	85.40	1.06	1077.93	269.48
FT	8	79.99	1.13	1150.81	143.85
FT	16	44.97	2.01	2046.97	127.94
FT	32	41.38	2.19	2224.35	69.51
IS	1	3.27	1.00	102.74	102.74
IS	2	4.64	0.70	72.30	36.15
IS	4	8.52	0.38	39.36	9.84
IS	8	9.07	0.36	37.01	4.63
IS	16	5.53	0.59	60.67	3.79
IS	32	6.28	0.52	53.41	1.67
LU	1	433.46	1.00	1150.80	1150.80
LU	2	182.47	2.38	2733.81	1366.90
LU	4	94.46	4.59	5280.61	1320.15
LU	8	57.91	7.49	8613.65	1076.71
LU	16	33.63	12.89	14830.88	926.93
LU	32	32.51	13.33	15342.58	479.46
MG	1	12.40	1.00	1569.20	1569.24
MG	2	7.04	1.76	2765.10	1382.56
MG	4	5.58	2.22	3486.50	871.62
MG	8	4.98	2.49	3911.00	488.87
MG	16	3.20	3.88	6080.60	380.03
MG	32	4.07	3.05	4779.10	149.35
SP	1	399.86	1.00	887.84	887.84
SP	4	131.83	3.03	2692.99	673.25
SP	9	101.47	3.94	3498.67	388.74
SP	16	77.29	5.17	4593.14	287.07

Figure 2: FT Benchmark on Lab Machines

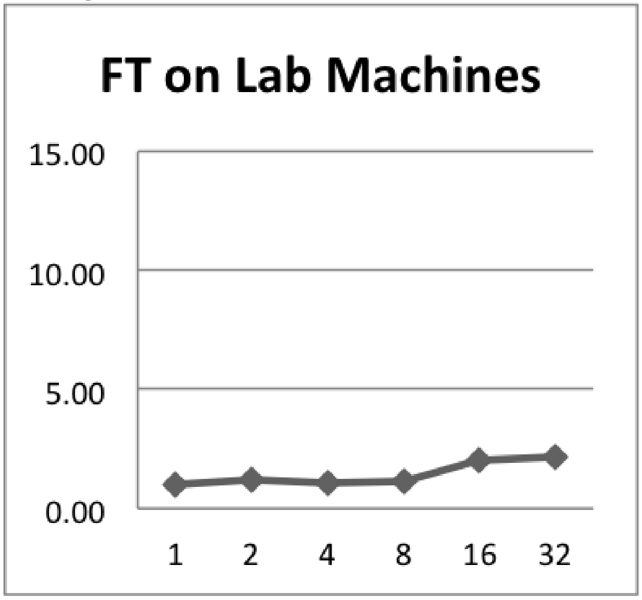


Figure 3: IS Benchmark on Lab Machines

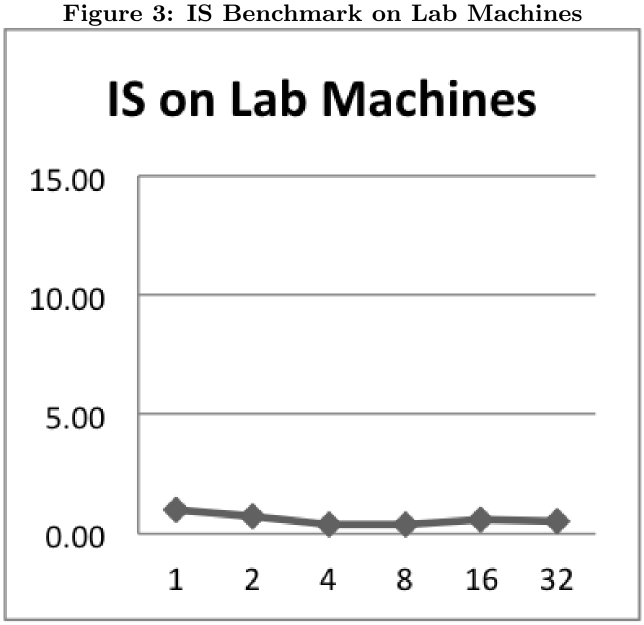


Figure 4: LU Benchmark on Lab Machines

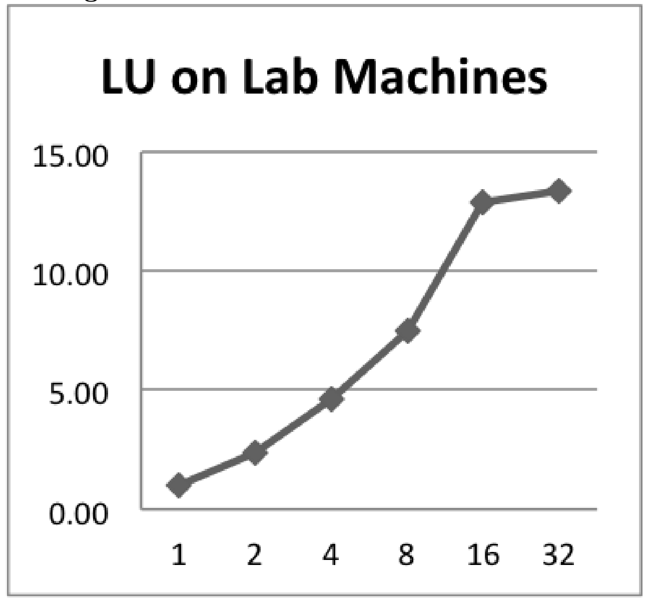


Figure 5: MG Benchmark on Lab Machines

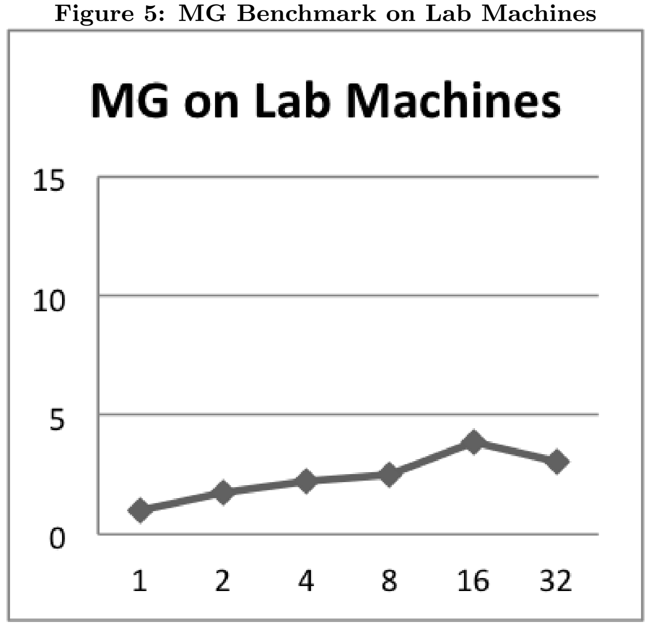


Figure 6: SP Benchmark on Lab Machines

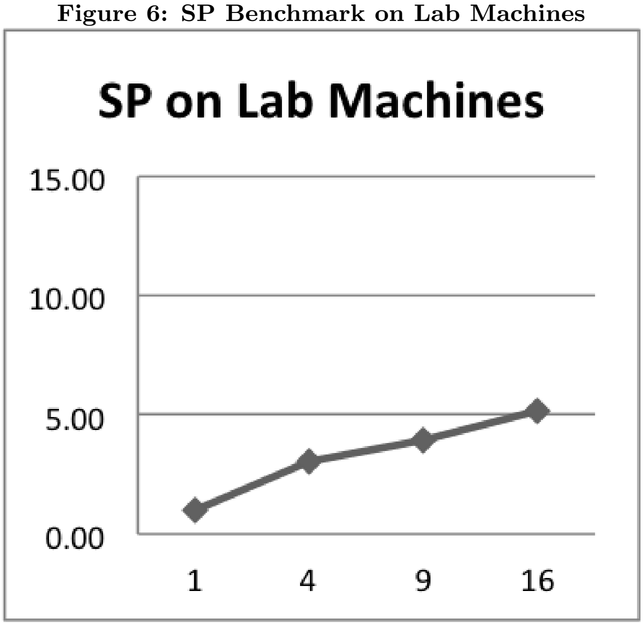
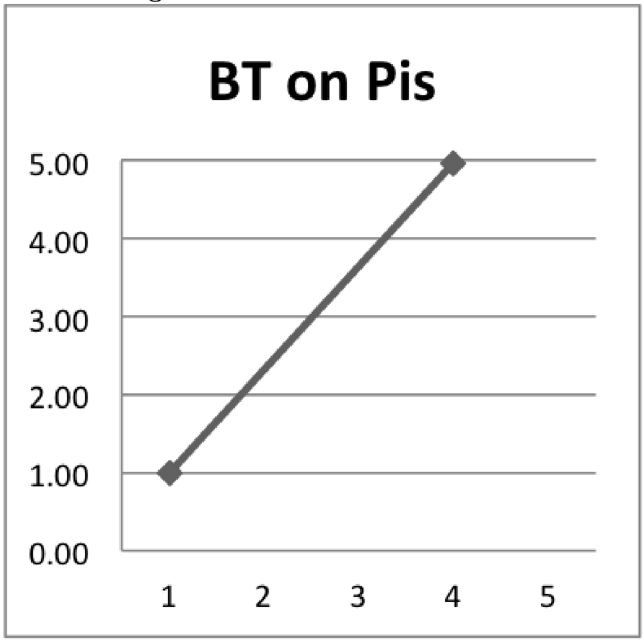


Figure 7: BT Benchmark on Pis



2.2 Raspberry Pis

Table 2 shows the Pis' NPB results. The class A benchmarks were used.

Figures 7 through 13 graph those speedup results.

3. ANALYSIS

Although it's not always the case, we typically notice a linear improvement of runtime when the number of nodes increases. This indicates that the NAS benchmarks, to some extent, are able to divide up their work to be performed on separate systems with their own memory and simply combine the results as needed.

Of course, the higher the speedup, the less relient the nodes are on each other. In some cases, we notice that the number of nodes is almost completely tied to the speedup, with perhaps only a small performance hit due to slight overhead. In other situations, there's a very significant performance hit when the workload is distributed (see Figures 3 and 10) and the speedup can even descrease. This happens when the distributed system has to spend more time per node dealing with the added overhead of initializing and message passing than it spends on the benchmark processing itself.

In a way, this benchmarking exercise serves to validate distrbuted computing in our eyes. Especially on the Raspberry Pis, despite being somewhat underwhelming on their own, we see that having them work together provides a huge increase in performance and that there's real value in using many unremarkable systems to process large amounts of data.

Figure 8: EP Benchmark on Pis

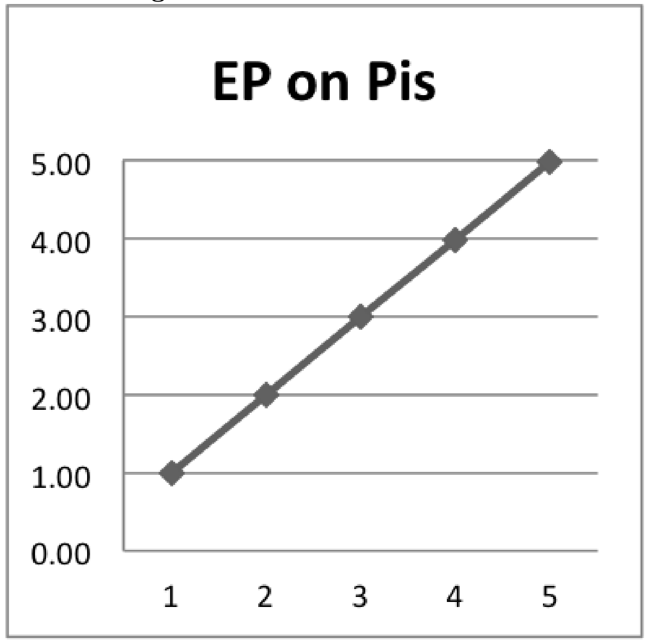


Figure 9: FT Benchmark on Pis

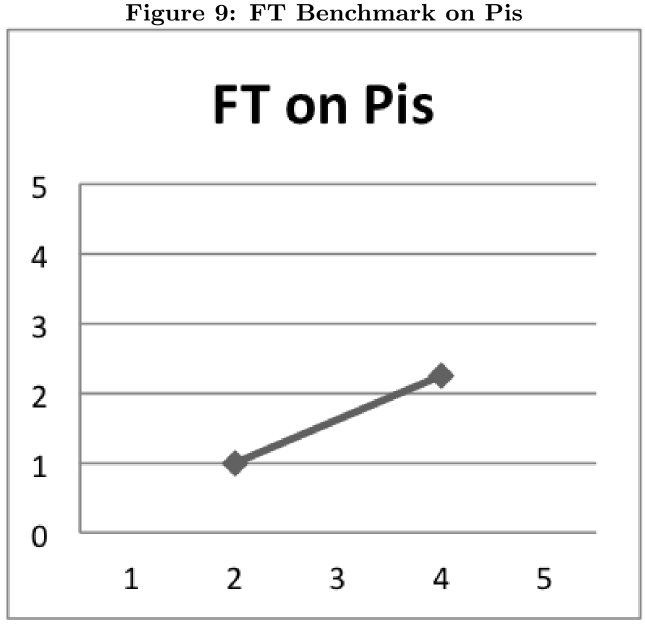


Table 2: Raspberry Pi Benchmark Results

Benchmark	Nodes	Time	Speedup	MOPS	Process MOPS
BT	1	4252.40	1.00	39.57	39.57
BT	4	858.28	4.95	196.07	49.02
EP	1	334.98	1.00	1.60	1.60
EP	2	167.65	2.00	3.20	1.60
EP	3	111.90	2.99	4.80	1.60
EP	4	84.16	3.98	6.38	1.59
EP	5	67.38	4.97	7.97	1.59
FT	2	211.66	1.00	33.72	16.86
FT	4	93.78	2.26	76.10	19.02
IS	1	24.78	1.00	3.39	3.39
IS	2	24.99	0.99	3.36	1.68
IS	4	16.07	1.54	5.22	1.31
LU	1	2507.90	1.00	47.57	47.57
LU	2	1297.50	1.93	91.95	45.97
LU	4	675.59	3.71	176.58	44.15
MG	2	98.53	1.00	39.50	19.75
MG	4	40.84	2.41	95.30	23.82
SP	1	3474.10	1.00	24.47	24.47
SP	4	938.64	3.70	90.57	22.64

Figure 10: IS Benchmark on Pis

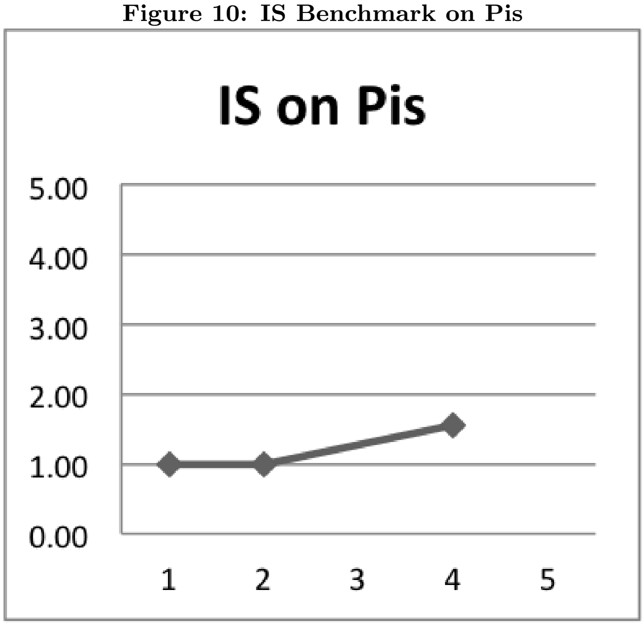


Figure 11: LU Benchmark on Pis

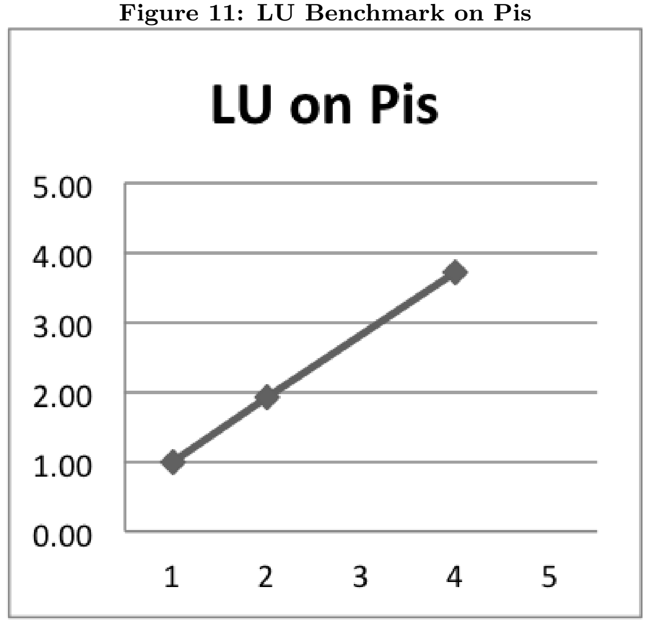


Figure 12: MG Benchmark on Pis

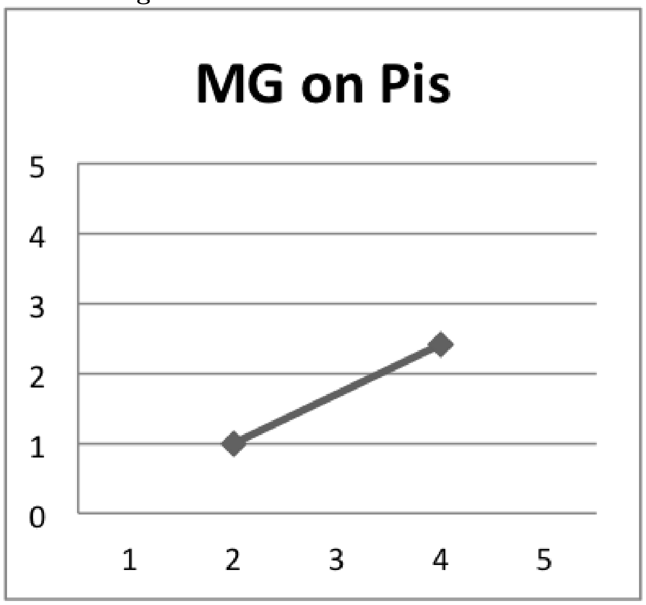


Figure 13: SP Benchmark on Pis