A Comprehensive Study of Characterizing Program Execution Time

Young-Kyoon Suh
yksuh@cs.arizona.edu

January 27, 2016

1 Experiment Notes

Task Length	Description	Time Length
PUT1∼PUT64	A regular PUT experiment with runs of 1000 samples (on `sodb12`).	2013-10-14 ∼ 2013-10-15
PUT128∼PUT2048	A regular PUT experiment with runs of 300 samples (on `sodb12`).	2013-12-12 ∼ 2013-12-21
PUT4096	A regular PUT experiment with a run of 300 samples (on `sodb12`).	2014-06-23 ∼ 2014-07-10
PUT8192	A regular PUT experiment with a run of 40 samples (on `sodb12`).	2015-04-23 ∼ 2015-04-27
PUT8192	A regular PUT experiment with a run of 260 samples (on `sodb12`).	2015-10-31 ∼ 2015-11-24
PUT16384	A regular PUT experiment with a run of 40 samples (on `sodb12`).	2015-04-23 ∼ 2015-04-23
PUT16384	A regular PUT experiment with a run of 260 samples (on `sodb12`).	2015-11-25 ∼ 2016-01-14

Table 1: Notes on the PUT data used for the histograms

Task Length	Description	Time Length
PUT1	A regular PUT experiment with a run of 20,000 samples on `sodb9`.	2015-12-15 ∼ 2015-12-15
PUT2	A Regular PUT experiment with a run of 20,000 samples on `sodb10`.	2015-12-15 ∼ 2015-12-15
PUT4096	A dual PUT experiment with a run of 500 samples on `sodb8`. Used `gettimeofday()` for measuring the elapsed time of each half of every PUT4096 (same for the following PUT2 and PUT64).	2015-11-08 ∼ 2015-12-25
PUT2	A dual PUT experiment with a run of 1,000 samples on `sodb9`.	2015-12-27 ∼ 2015-12-27
PUT64	A dual PUT experiment with a run of 1,000 samples on `sodb10`	2015-12-27 ∼ 2015-12-27

Table 2: Notes on the new PUT experiments

2 Summary of the EMPv4 data

	Num. of Samples	Minimum (msec)	Maximum (msec)	Average (msec)	Std. Dev. (msec)
PUT1	1,000	999.0	1,005.0	1,002.4	0.73
PUT2	1,000	1,996.0	2,007.0	2,004.5	1.38
PUT4	1,000	4,004.0	4,012.0	4,008.6	1.64
PUT8	1,000	8,014.0	8,023.0	8,018.1	1.72
PUT16	1,000	16,029.0	16,041.0	16,034.3	1.86
PUT32	1,000	32,064.0	32,084.0	32,068.2	2.05
PUT64	1,000	64,129.0	64,145.0	64,135.0	2.27
PUT128	300	128,244.0	128,260.0	128,251.2	2.32
PUT256	300	256,494.0	256,523.0	256,502.3	3.29
PUT512	300	512,995.0	513,152.0	513,005.1	9.41
PUT1024	300	1,025,997.0	1,026,141.0	1,026,012.4	11.43
PUT2048	300	2,051,981.0	2,052,156.0	2,052,012.0	11.19
PUT4096	300	4,105,451.0	4,105,629.0	4,105,526.0	25.98
PUT8192	40 (last Apr)	8,207,870.0	8,207,967.0	8,207,918.0	21.03
PUT8192	260 (Nov)	8,210,940.0	8,211,196.0	8,211,049.0	36.60
PUT16384	40 (last Apr)	16,415,757.0	16,415,964.0	16,415,810.3	40.43
PUT16384	260 (Nov)	16,422,028	16,422,389	16,422,153.0	52.54

Table 3: PT statistics by EMPv4 (extension of Table VI in the EMP paper)

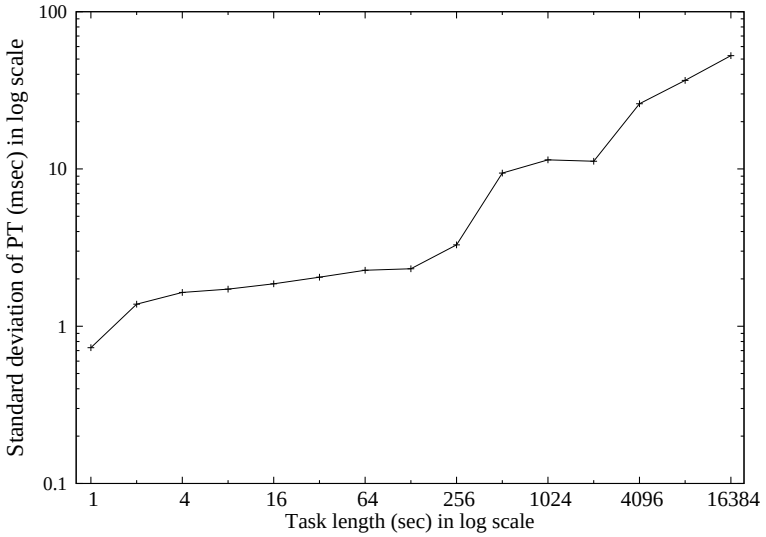
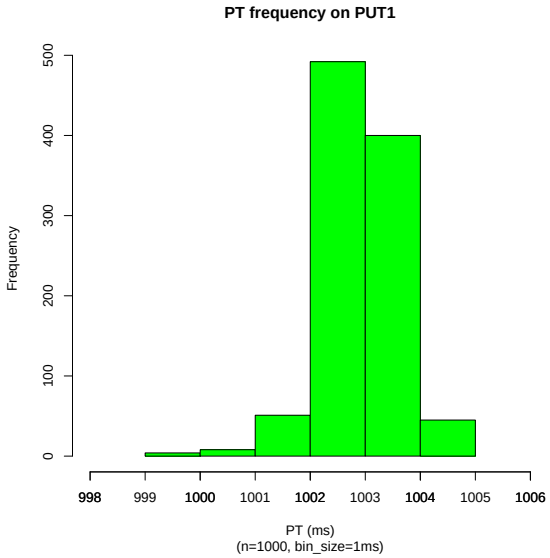
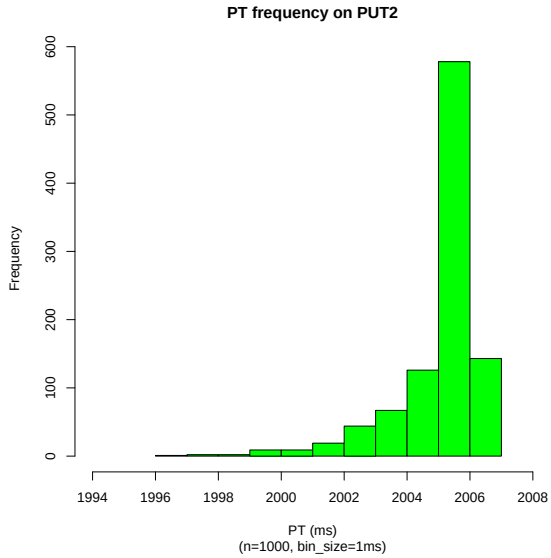


Figure 1: Std. dev. of PT over increasing task length

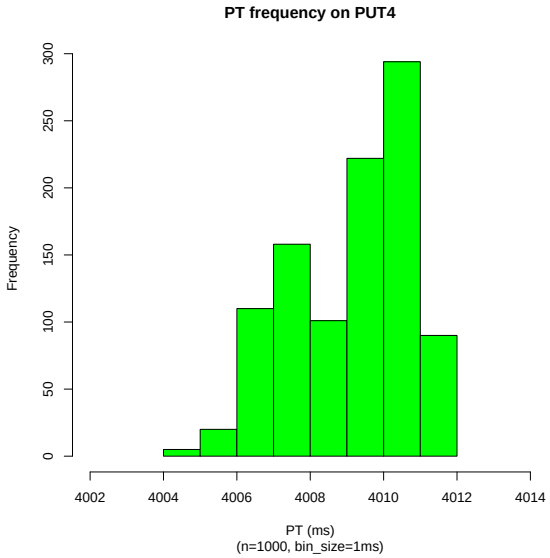
3 Histograms on the EMPv4 Data



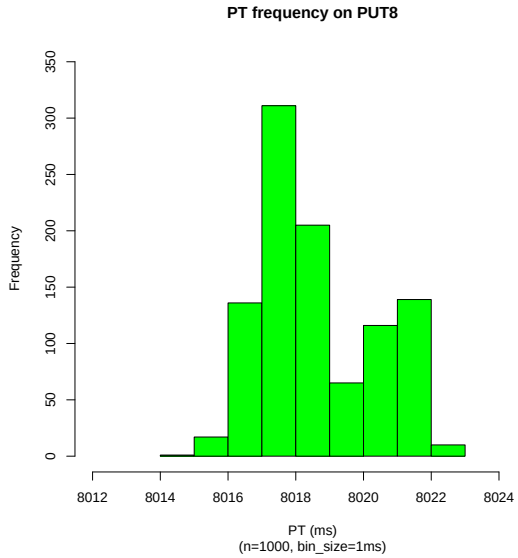
(a) PT frequency on PUT1



(b) PT frequency on PUT2

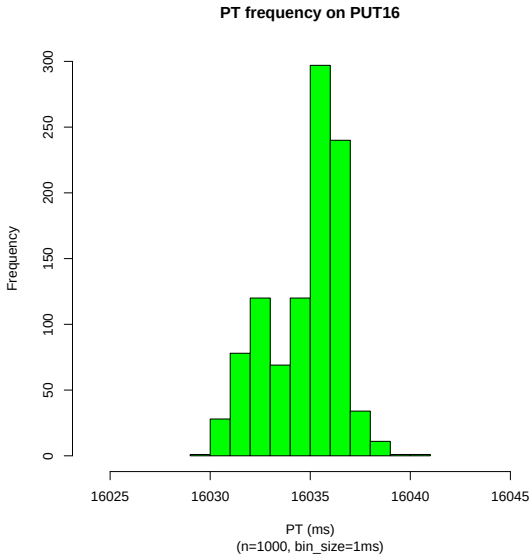


(c) PT frequency on PUT4

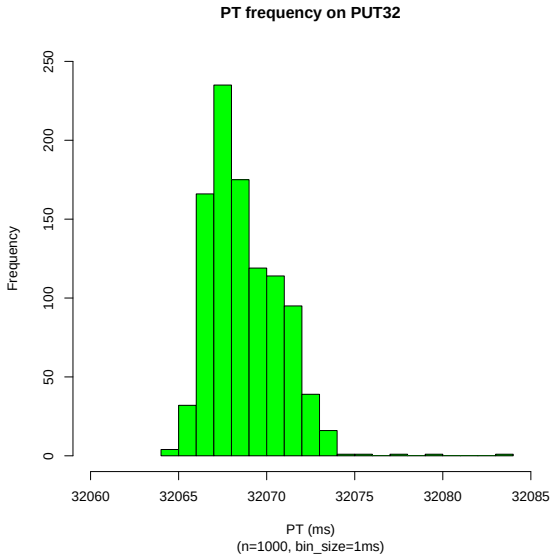


(d) PT frequency on PUT8

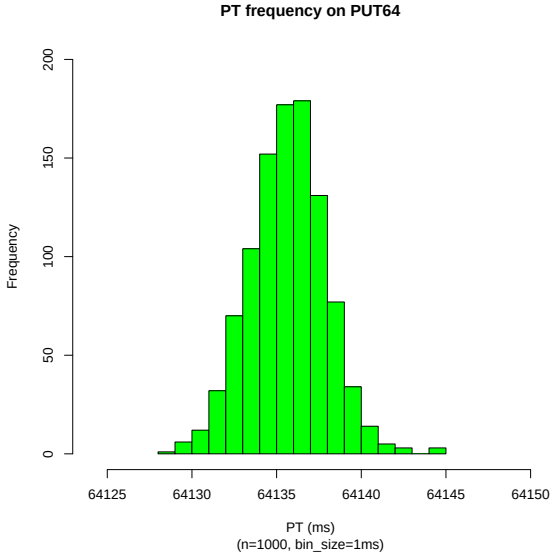
Figure 2: PT Histograms of PUT1 ... PUT8



(a) PT frequency on PUT16

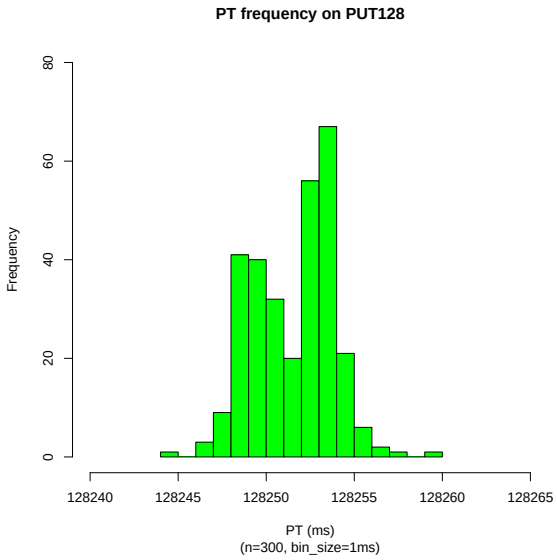


(b) PT frequency on PUT32

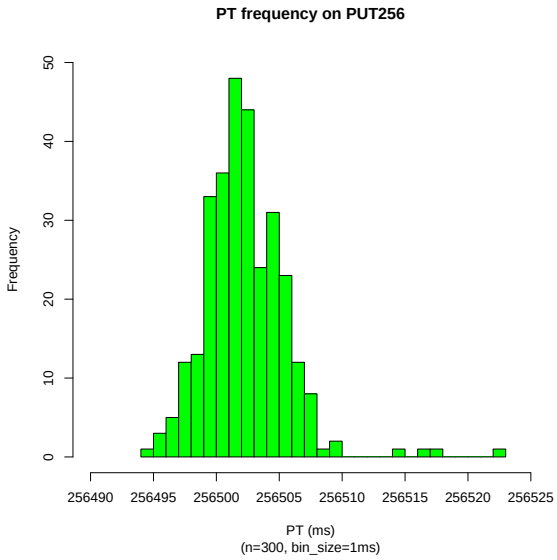


(c) PT frequency on PUT64

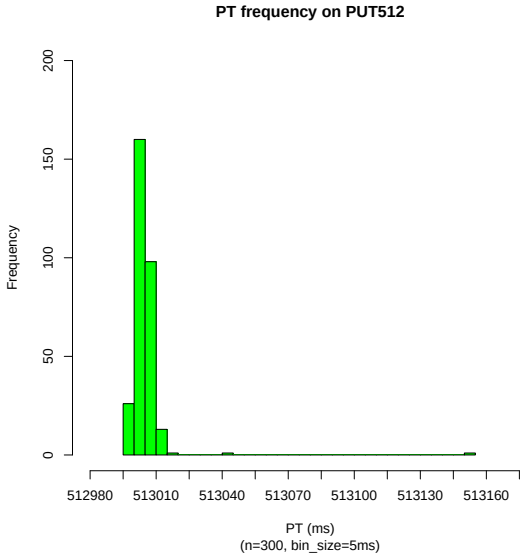
Figure 3: PT Histograms of PUT16 ... PUT64



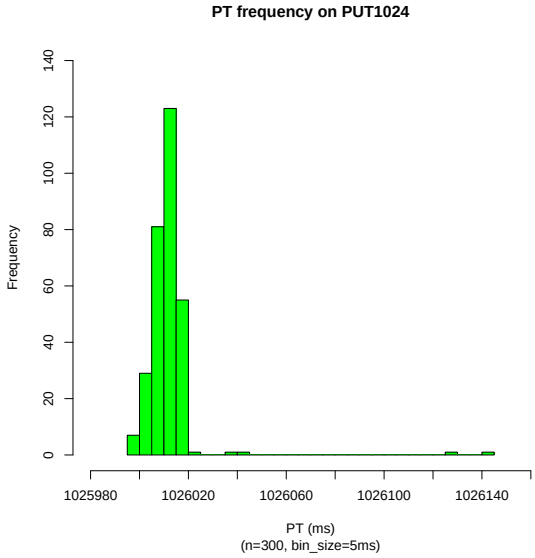
(a) PT frequency on PUT128



(b) PT frequency on PUT256

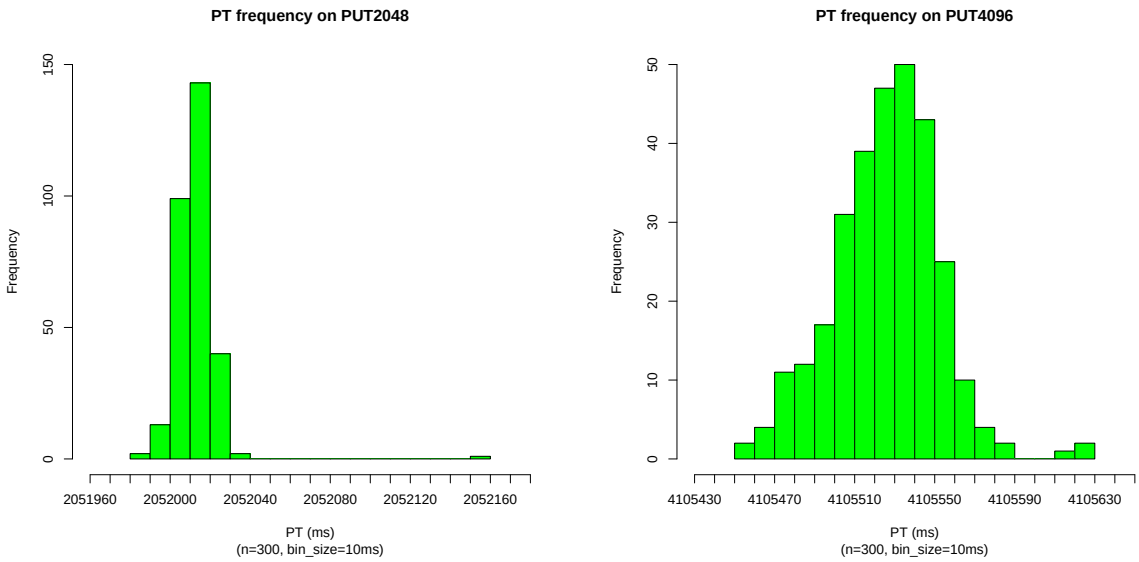


(c) PT frequency on PUT512



(d) PT frequency on PUT1024

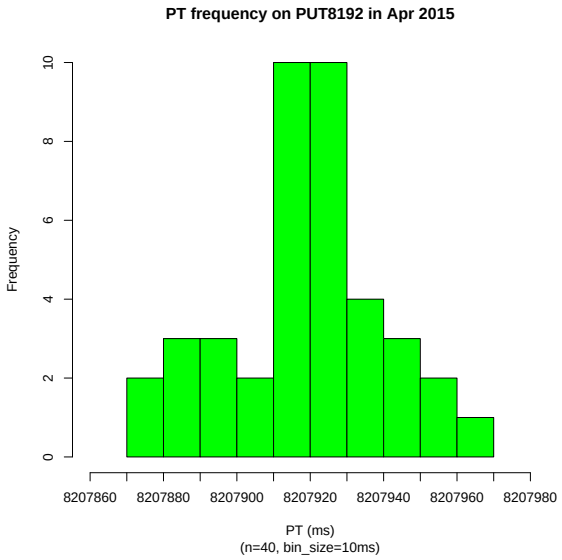
Figure 4: PT Histograms of PUT128 ... PUT1024



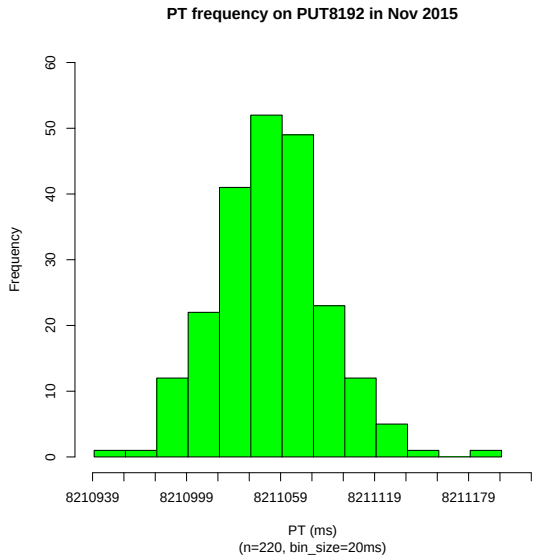
(a) PT frequency on PUT2048

(b) PT frequency on PUT4096

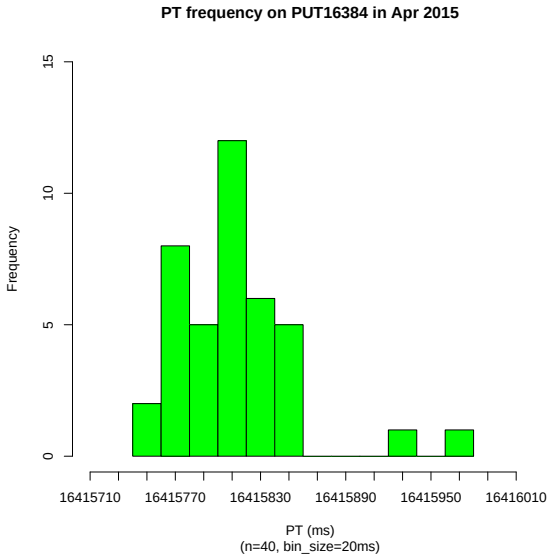
Figure 5: PT Histograms of PUT2048 and PUT4096



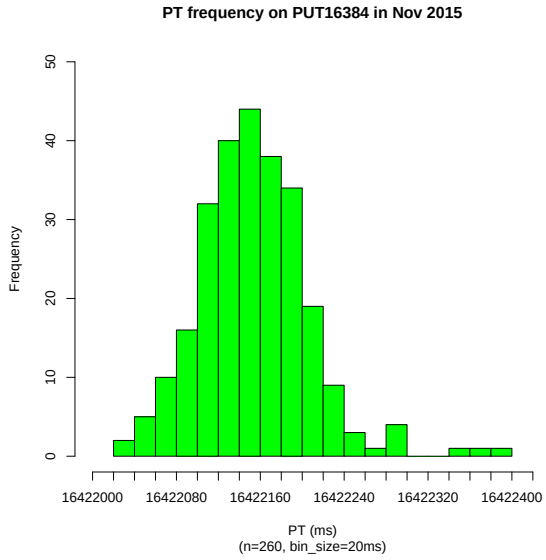
(a) PT frequency on PUT8192 with 40 samples (See Table 1.)



(b) PT frequency on PUT8192 with 260 samples (See Table 1.)



(c) PT frequency on PUT16384 with 40 samples (See Table 1.)

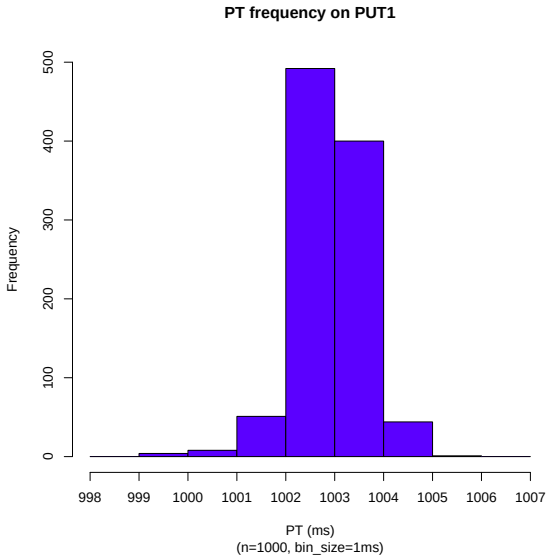


(d) PT frequency on PUT16384 with 260 samples (See Table 1.)

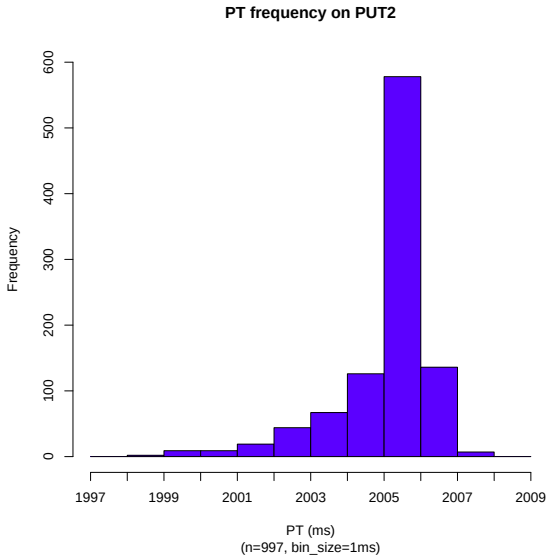
Figure 6: PT Histograms of PUT8192 and PUT16384

4 Histograms on the EMPv5 Data

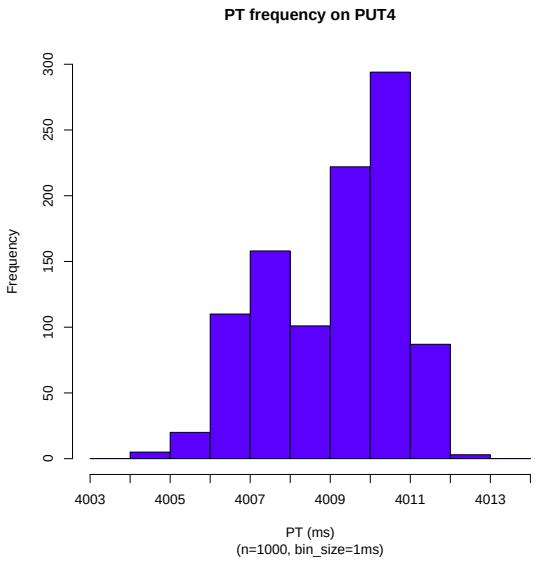
EMPv5 removes the outliers above and below the average ± *five* standard deviations of program times measured by EMPv4.



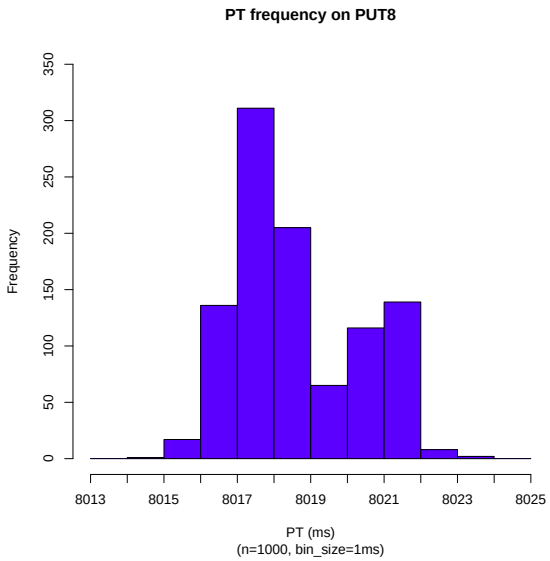
(a) PT frequency on PUT1



(b) PT frequency on PUT2

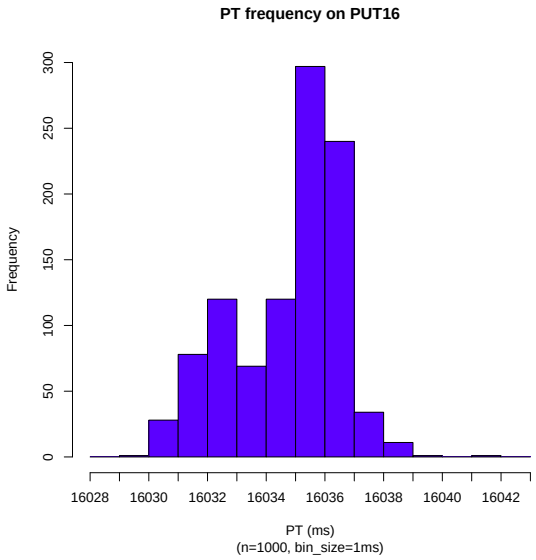


(c) PT frequency on PUT4

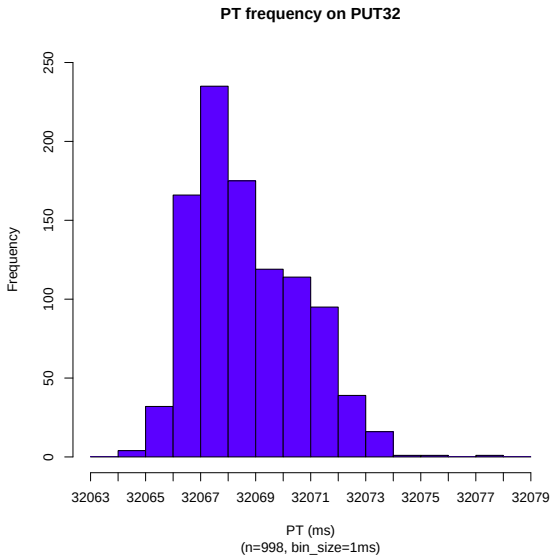


(d) PT frequency on PUT8

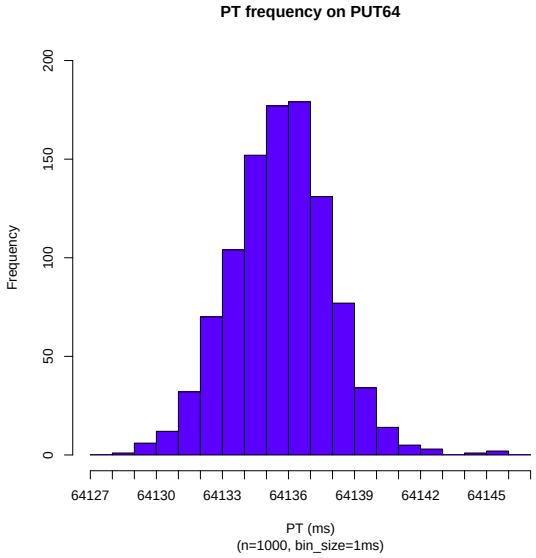
Figure 7: PT Histograms of PUT1 ... PUT8



(a) PT frequency on PUT16

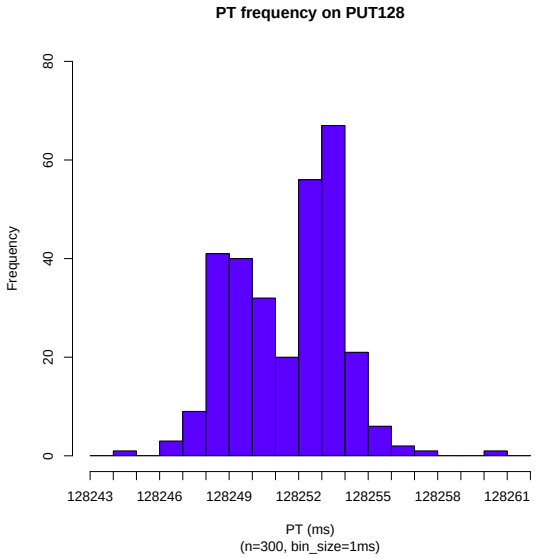


(b) PT frequency on PUT32

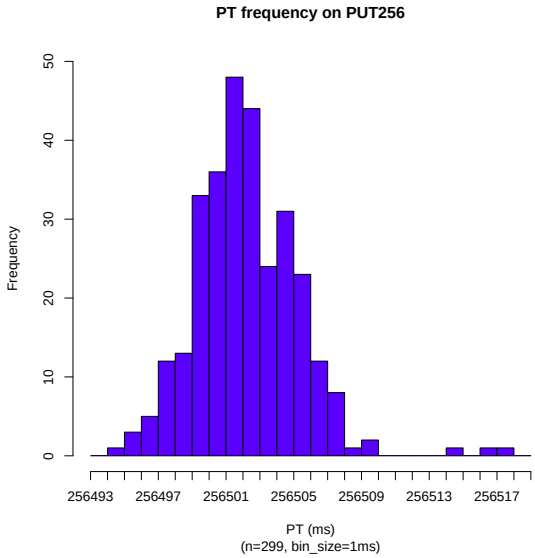


(c) PT frequency on PUT64

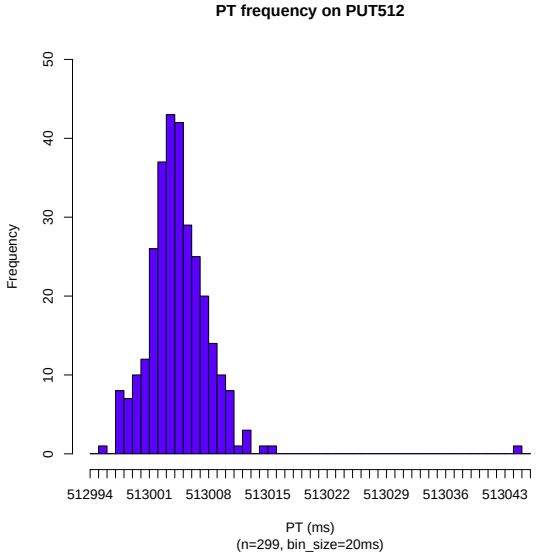
Figure 8: PT Histograms of PUT16 ... PUT64



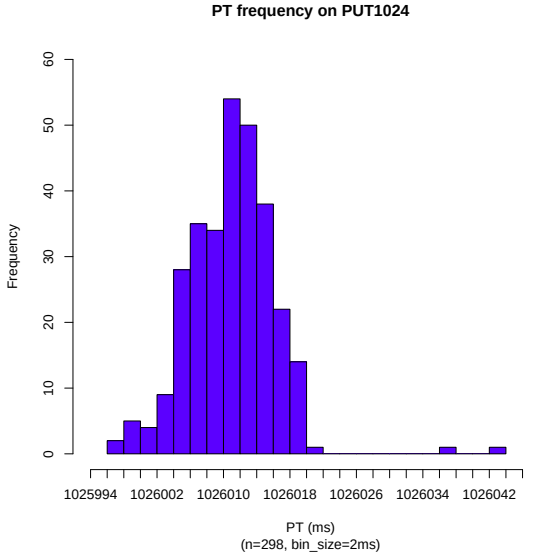
(a) PT frequency on PUT128



(b) PT frequency on PUT256

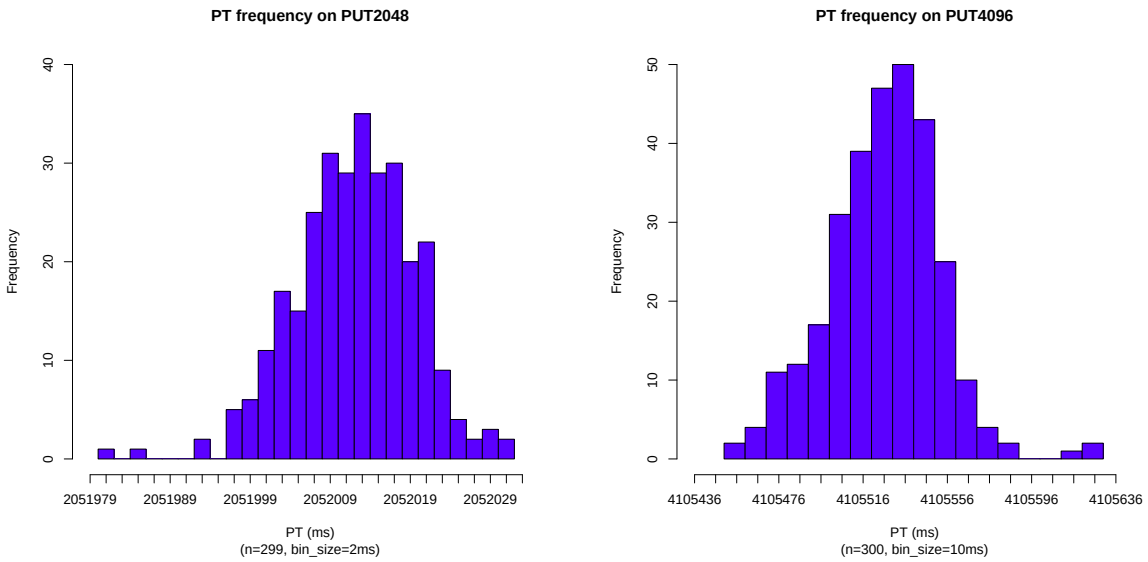


(c) PT frequency on PUT512



(d) PT frequency on PUT1024

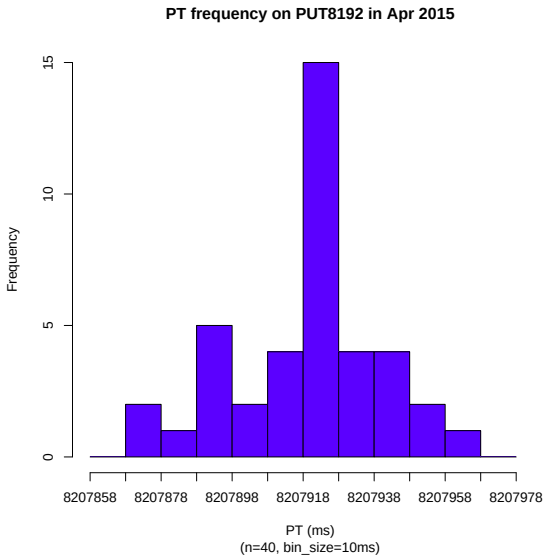
Figure 9: PT Histograms of PUT128 ... PUT1024



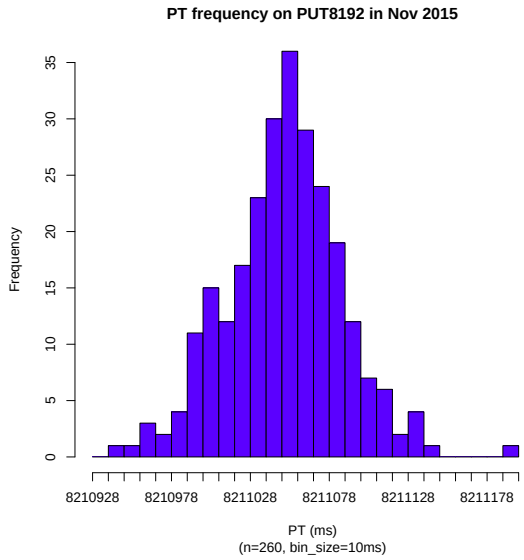
(a) PT frequency on PUT2048

(b) PT frequency on PUT4096

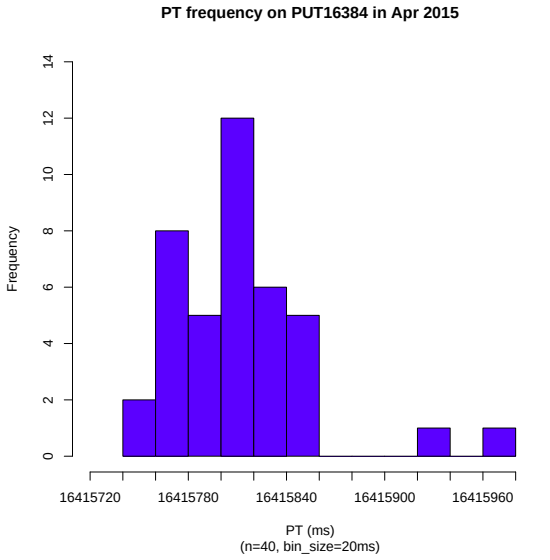
Figure 10: PT Histograms of PUT2048 and PUT4096



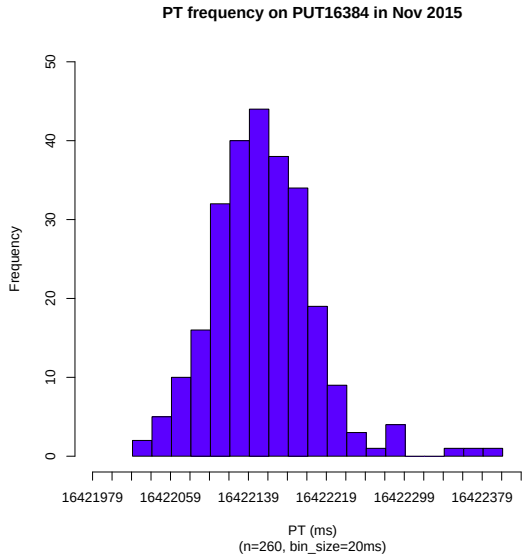
(a) PT frequency on PUT8192 with 40 samples (See Table 1.)



(b) PT frequency on PUT8192 with 260 samples (See Table 1.)



(c) PT frequency on PUT16384 with 40 samples (See Table 1.)



(d) PT frequency on PUT16384 with 260 samples (See Table 1.))

Figure 11: PT Histograms of PUT8192 and PUT16384

5 Sample Size vs. Standard Deviation of PT

Num. of Samples	Std. Dev. (msec)	
	PUT1	PUT2
1,000	1.07	1.40
2,000	1.06	1.39
3,000	1.07	1.38
4,000	1.07	1.37
5,000	1.07	1.40
6,000	1.06	1.70
7,000	1.06	1.65
8,000	1.07	1.62
9,000	1.07	1.60
10,000	1.07	1.58
11,000	1.08	1.57
12,000	1.08	1.56
13,000	1.08	1.54
14,000	1.08	1.53
15,000	1.08	1.52
16,000	1.08	1.51
17,000	1.08	1.50
18,000	1.08	1.50
19,000	1.08	1.50
20,000	1.08	1.49

Table 4: Std. Dev. of PUT1 and PUT2 over increasing sample size

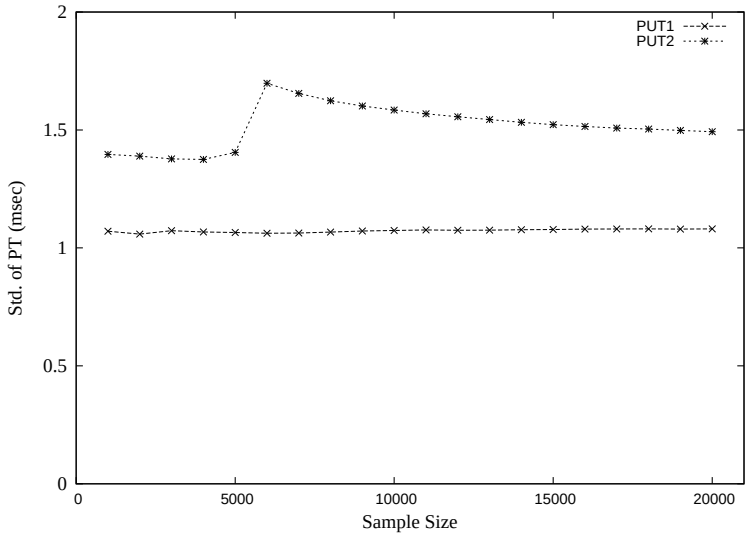


Figure 12: Std. dev. of PT on PUT1 and PUT2 over increasing sample size

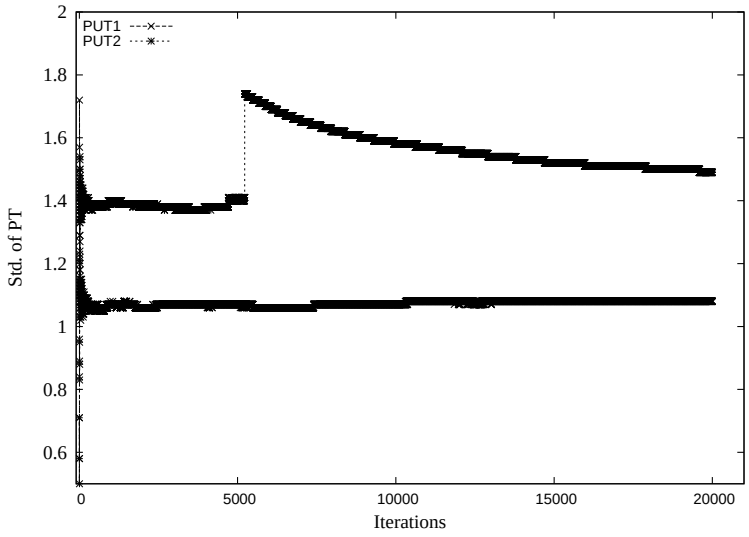
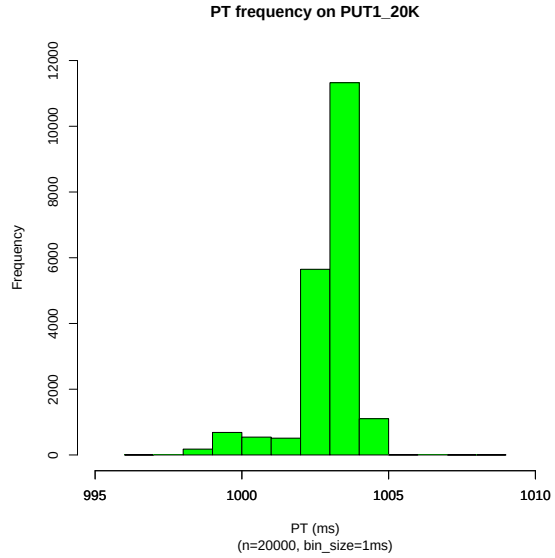


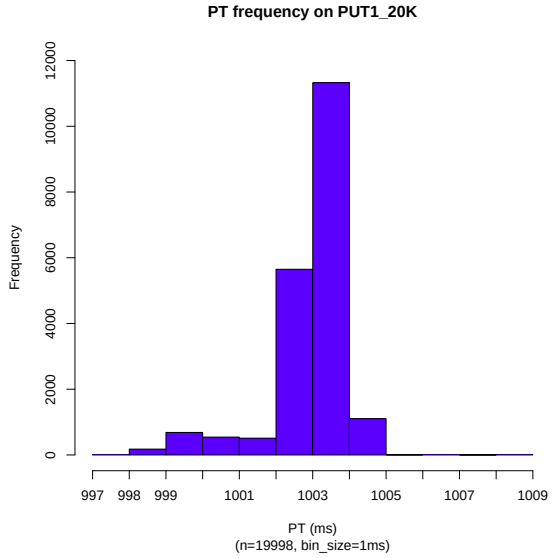
Figure 13: Std. dev. of PT on PUT1 and PUT2 over increasing sample size

PUT2	Program Time
`incr_work`	2078 msecs (at the 5276th iteration)
Daemon Processes	Program Time
`md0_raid1`	1 msec
`proc_monitor`	198 msecs
`rhn_check`	460 msecs
Total	659 msecs

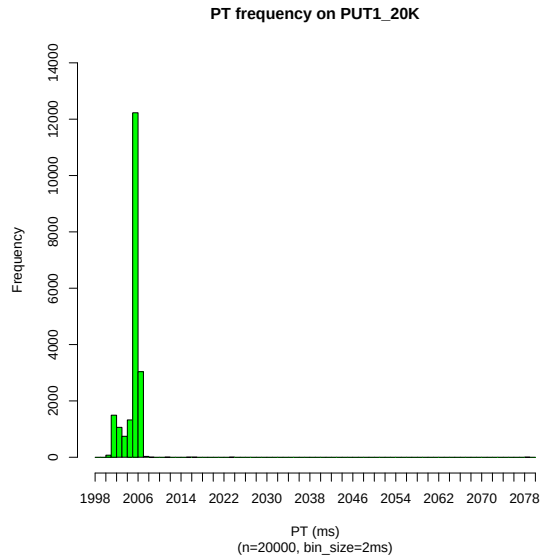
Table 5: The daemon processes captured at the hike of PUT2



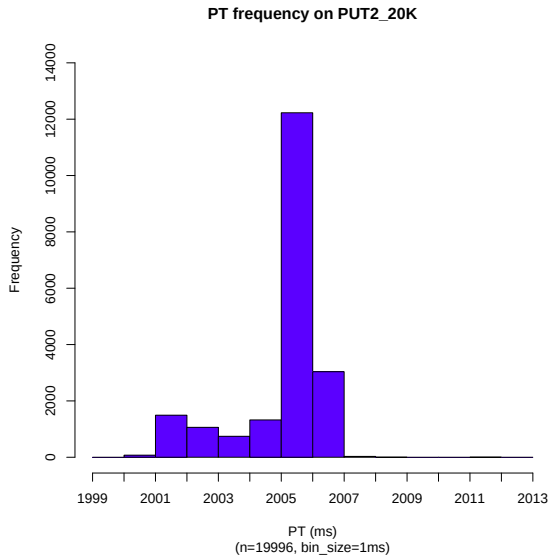
(a) PT frequency on PUT1 with 20,000 samples (See Table 2.)



(b) PT frequency on PUT1 excluding the outliers out of the 20,000 samples (See Table 2.)



(c) PT frequency on PUT2 with 20,000 samples (See Table 2.)



(d) PT frequency on PUT2 excluding the outliers out of the 20,000 samples (See Table 2.)

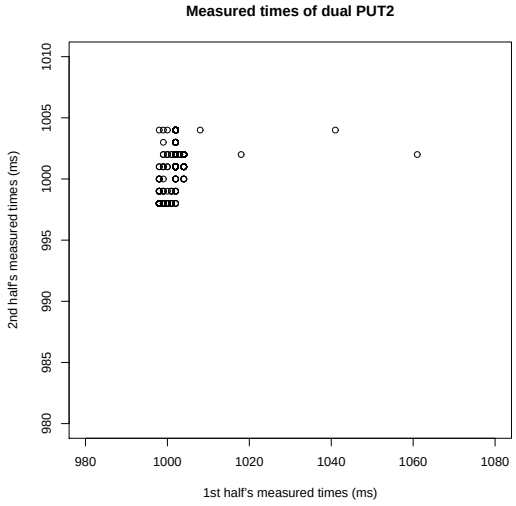
Figure 14: PT Histograms of PUT1 and PUT2 by 20,000 trials

6 Dual PUT Experiments

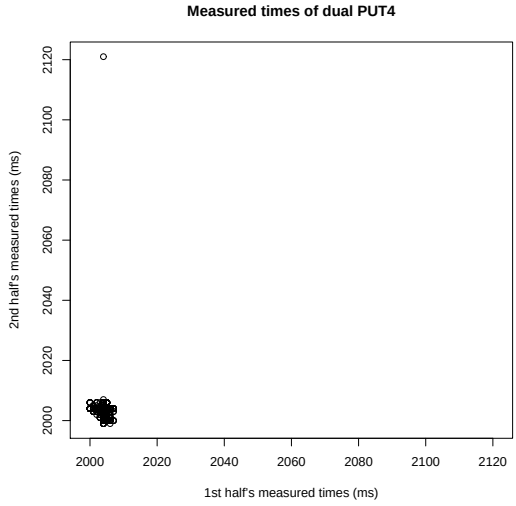
6.1 Scatter Plots

Dual PUT	PUT2	PUT4	PUT8	PUT16	PUT32	PUT64	PUT4096
Sample Size	1000	1000	1000	1000	1000	1000	500
Correlation Coefficients	0.3	-0.07 (-0.15 except max)	0.8	0.3	-0.01	-0.01	-0.01

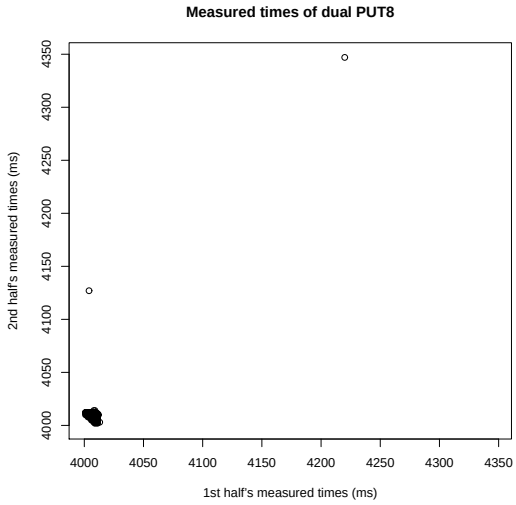
Table 6: Correlation Coefficients of Dual PUT Experiments



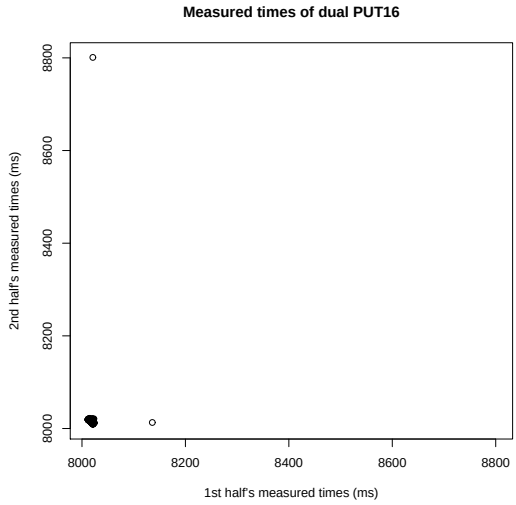
(a) Measured times on dual PUT2



(b) Measured times on dual PUT4

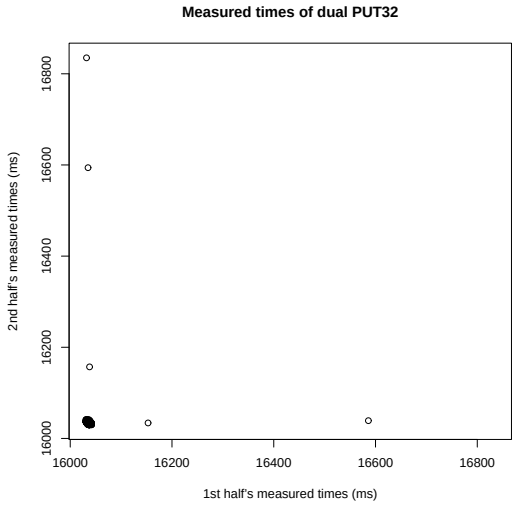


(c) Measured times on dual PUT8

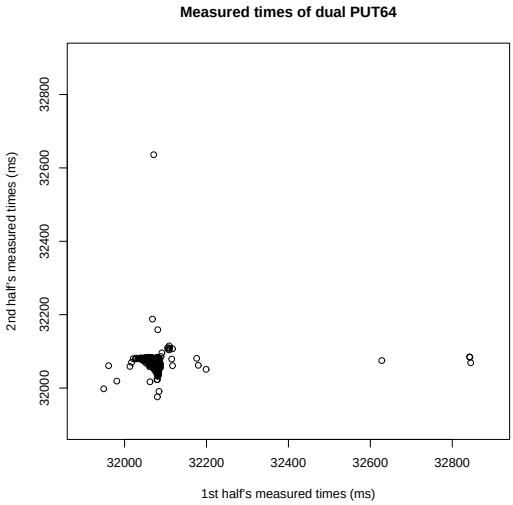


(d) Measured times on dual PUT16

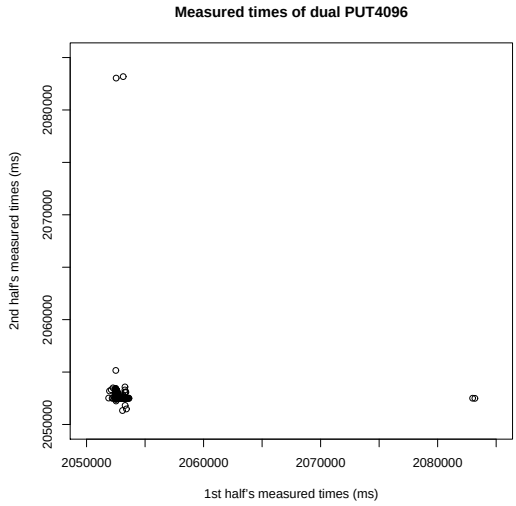
Figure 15: Scatter plots on dual PUT2∼PUT16



(a) Measured times on dual PUT32



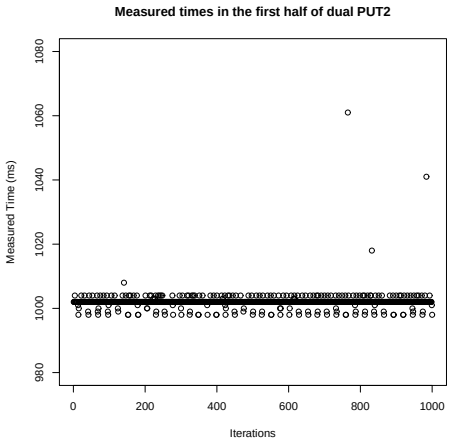
(b) Measured times on dual PUT64



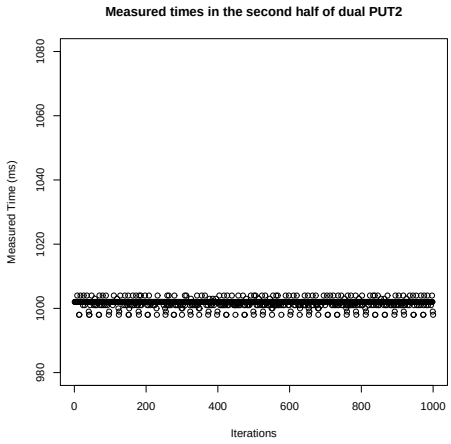
(c) Measured times on dual PUT4096

Figure 16: Scatter plots on dual PUT32∼PUT4096

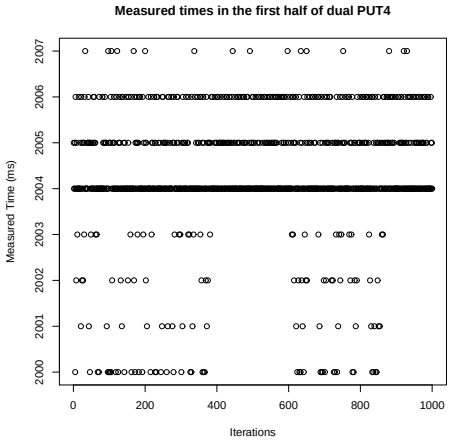
6.2 Program Time Comparison



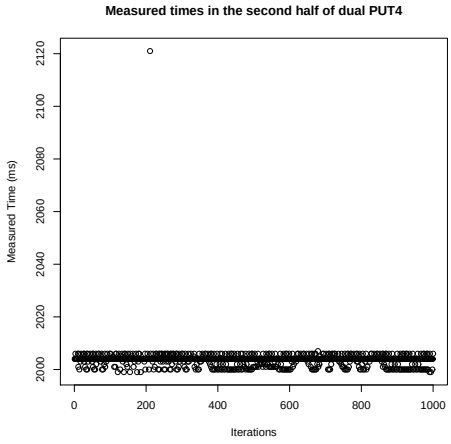
(a) 1st half's measured times on dual PUT2



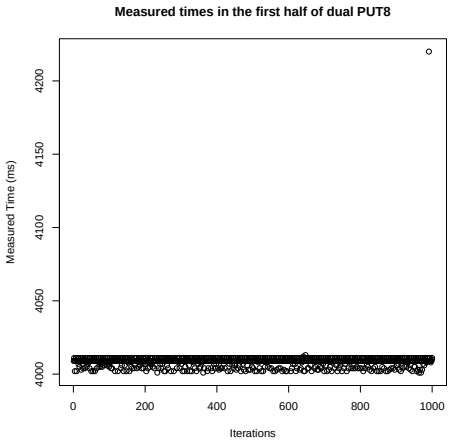
(b) 2nd half's measured times on dual PUT2



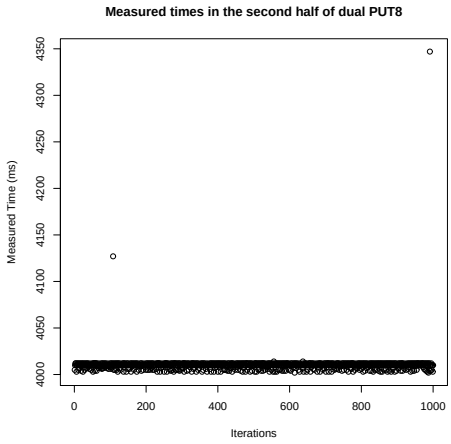
(c) 1st half's measured times on dual PUT4



(d) 2nd half's measured times on dual PUT4

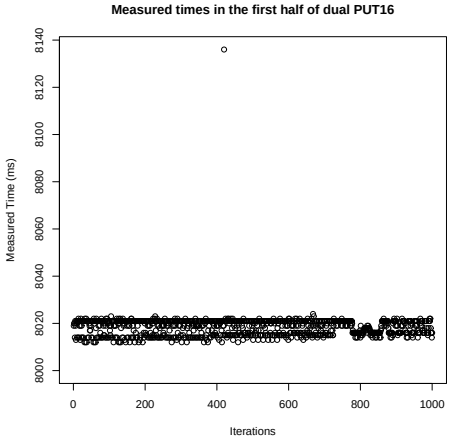


(e) 1st half's measured times on dual PUT8

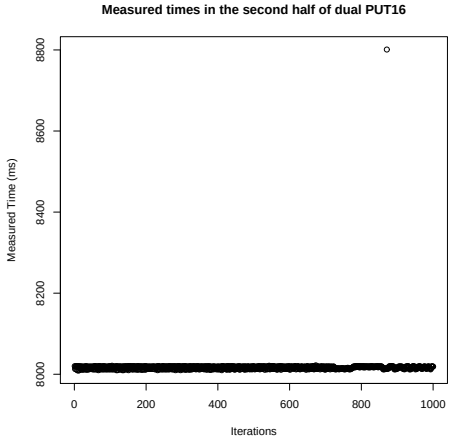


(f) 2nd half's measured times on dual PUT8

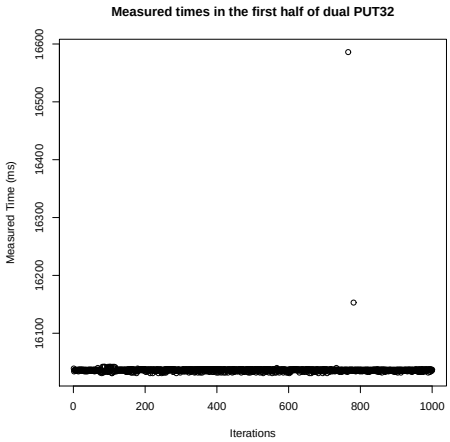
Figure 17: Measured time comparison on dual PUT2∼PUT8



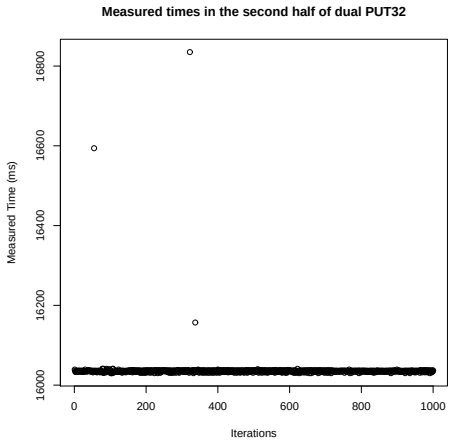
(a) 1st half's measured times on dual PUT16



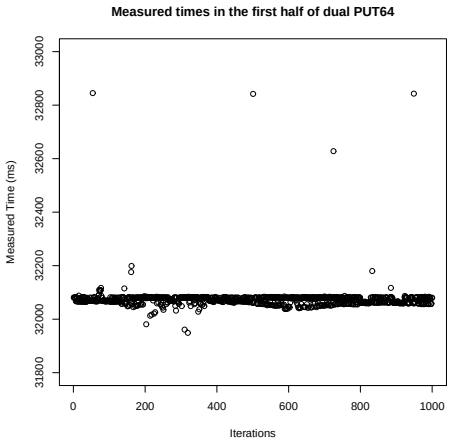
(b) 2nd half's measured times on dual PUT16



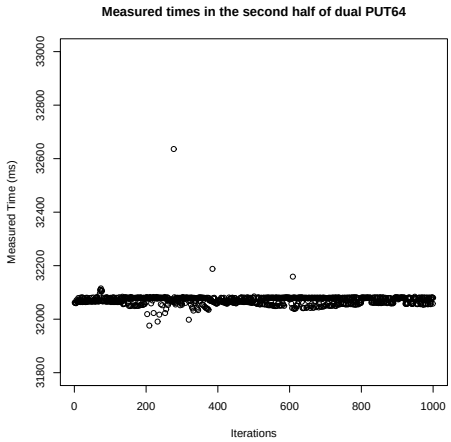
(c) 1st half's measured times on dual PUT32



(d) 2nd half's measured times on dual PUT32

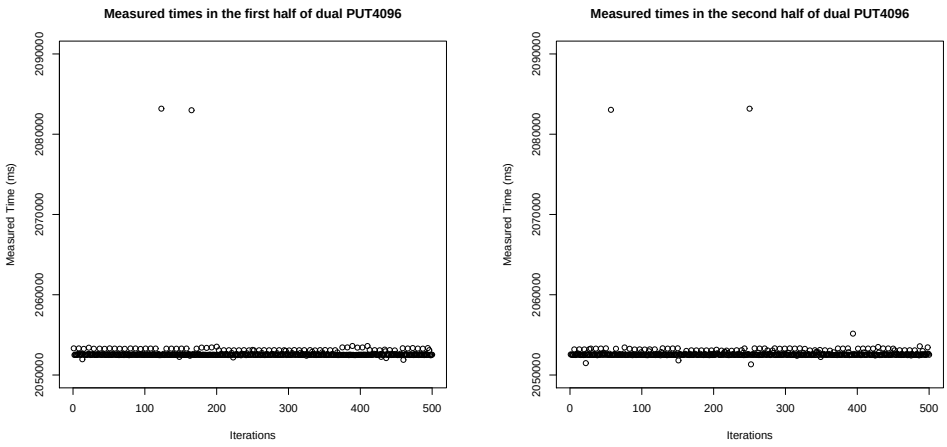


(e) 1st half's measured times on dual PUT64



(f) 2nd half's measured times on dual PUT64

Figure 18: Measured time comparison on dual PUT16∼PUT64

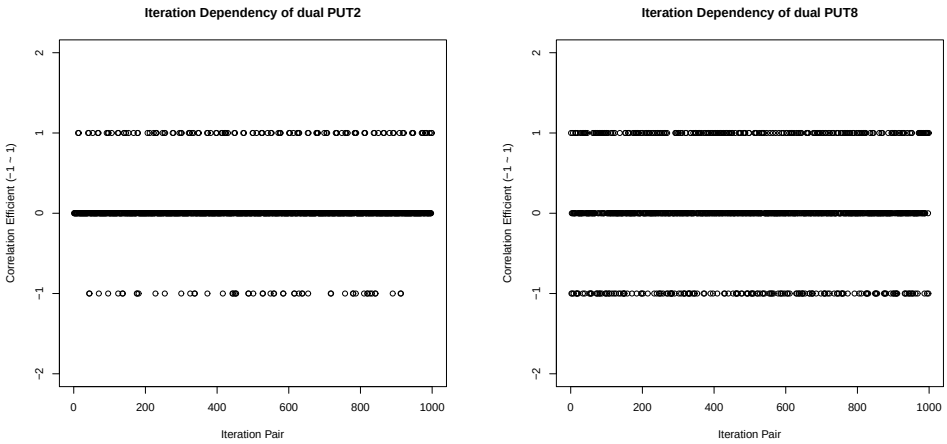


(a) 1st half's measured times on dual PUT4096 (b) 2nd half's measured times on dual PUT4096

Figure 19: Measured time comparison on dual PUT4096

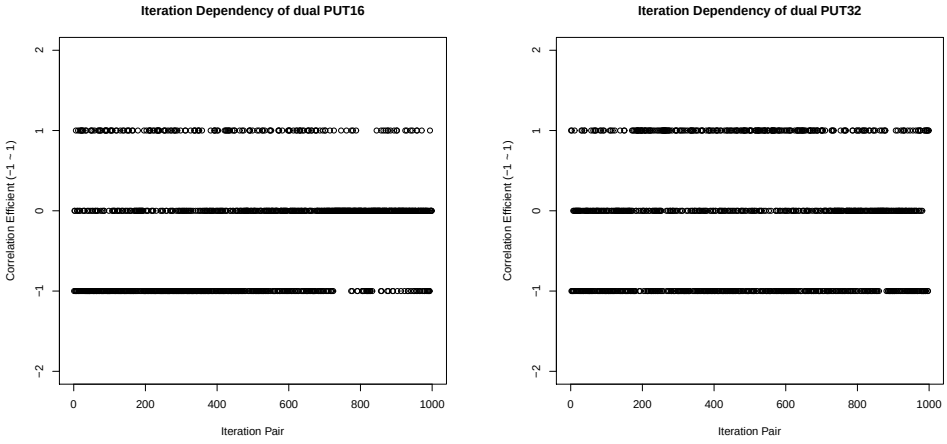
6.3 Successive Iterations' Dependency

Given a pair of successive iterations, we compute the correlation efficient of the pair. E.g., given dual PUT2, the first half's and second half's measured times were (999, 1001) at iteration 974 and (1002, 1004) at iteration 975. The computed correlation efficient of this pair was 1. (command: `cor(c(999, 1002), c(1001, 1004))`) On the other hand, the first half's and second half's measured times were (1002, 1001) at iteration 965 and (1002, 1004) at iteration 966. The correlation of efficient of this pair is -1. But the correlation efficient is 0 for the following pair: (1002, 1002) at iteration 102 and (1002, 1002) at iteration 103.



(a) Scatter plot on dual PUT2

(b) Scatter plot on dual PUT8



(c) Scatter plot on dual PUT16

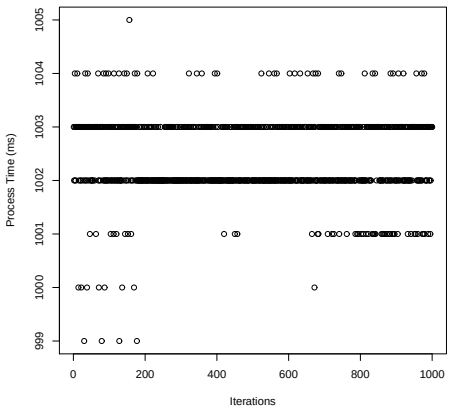
(d) Scatter plot on dual PUT32

Figure 20: Iteration Dependency on dual PUT2∼PUT32

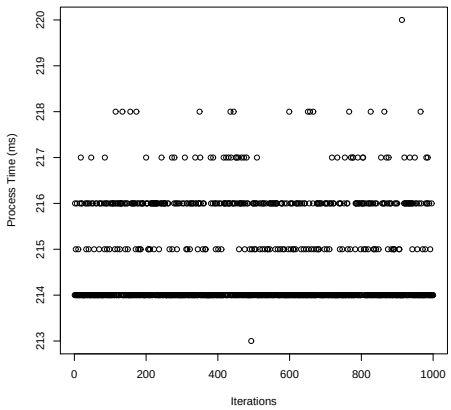
7 Correlations between the Program Times of Daemon Processes and PUT

PUT	Correlation Coefficient (EMPv4)	Correlation Coefficient (EMPv5)
PUT1	-0.2	-0.2
PUT2	-0.005	-0.009
PUT4	-0.05	-0.05
PUT8	0.1	0.1
PUT16	0.1	0.1
PUT32	0.3	0.15
PUT64	0.2	0.2
PUT128	0.2	0.2
PUT256	0.4	0.4
PUT512	0.9	0.6
PUT1024	0.9	0.2
PUT2048	0.8	0.24
PUT4096	0.4	0.4
PUT8192 in Apr	0.4	0.4
PUT8192 in Nov	0.3	0.3
PUT16384 in Apr	0.4	0.8
PUT16384 in Nov	0.5	0.5

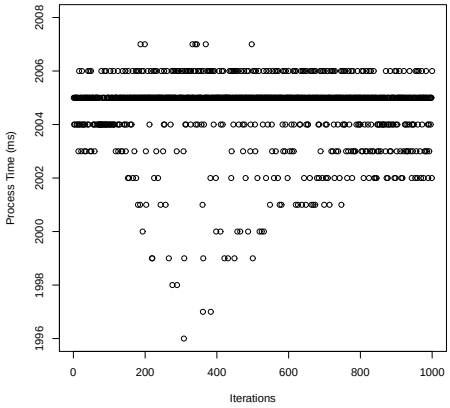
Table 7: Correlation Coefficients between Program Times of Daemon and PUT by EMPv4 and EMPv5



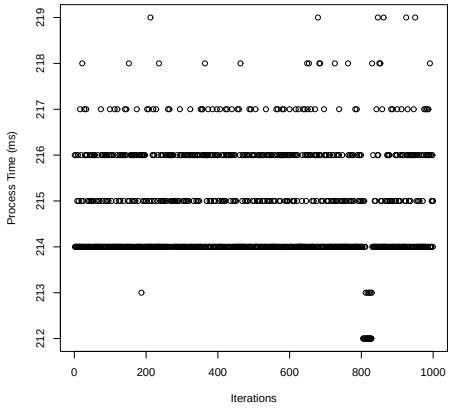
(a) PUT1's PTs



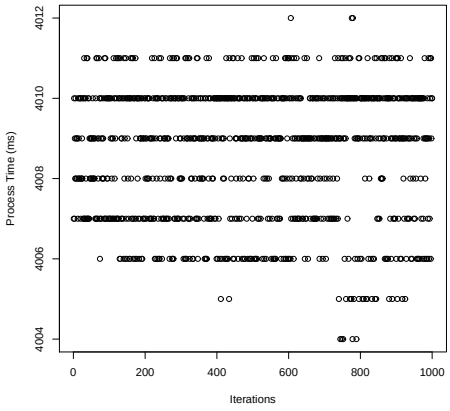
(b) PUT1's daemon PTs



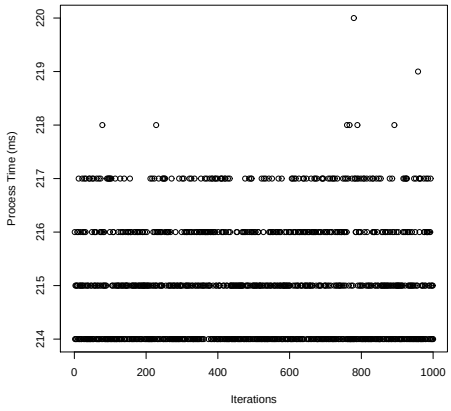
(c) PUT2's PTs



(d) PUT2's daemon PTs

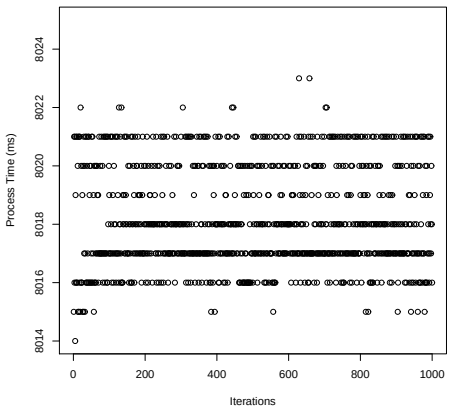


(e) PUT4's PTs

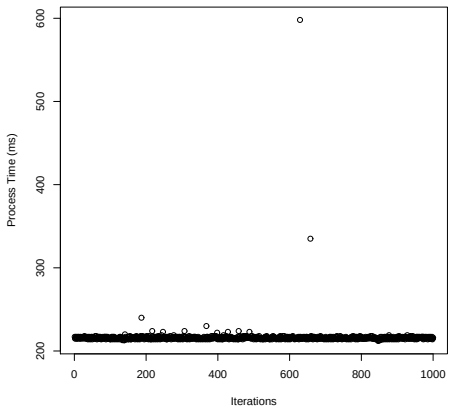


(f) PUT4's daemon PTs

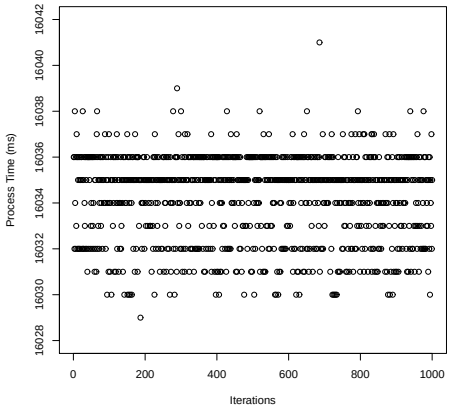
Figure 21: Program times between PUT1∼PUT4 vs. Daemon processes



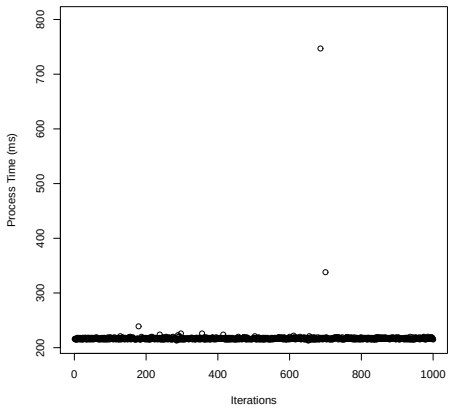
(a) PUT8's PTs



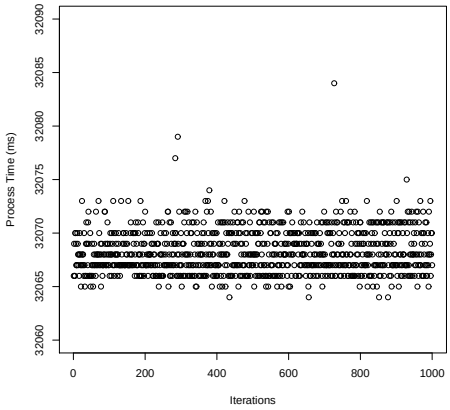
(b) PUT8's daemon PTs



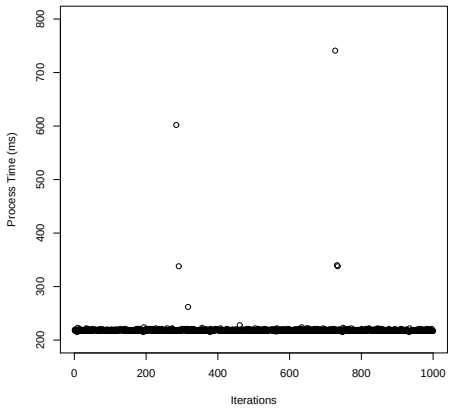
(c) PUT16's PTs



(d) PUT16's daemon PTs

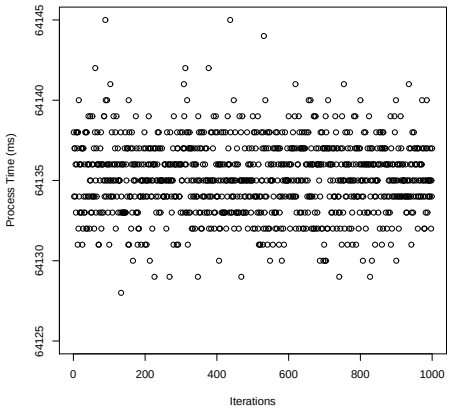


(e) PUT32's PTs

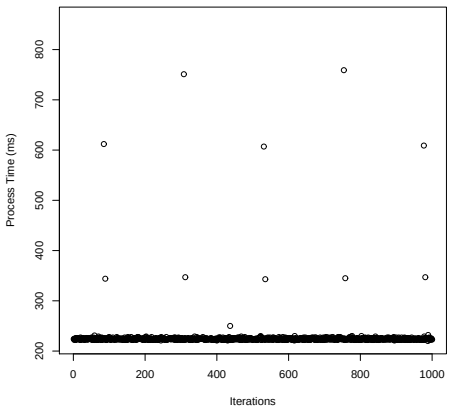


(f) PUT32's daemon PTs

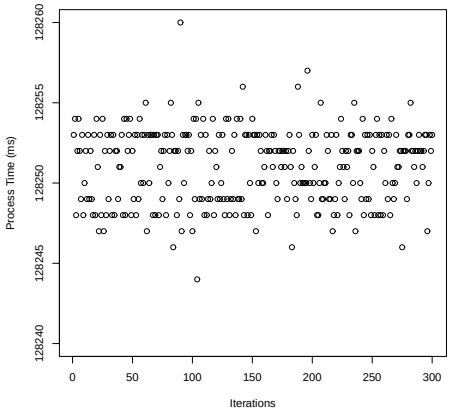
Figure 22: Program times between PUT8∼PUT32 vs. Daemon processes



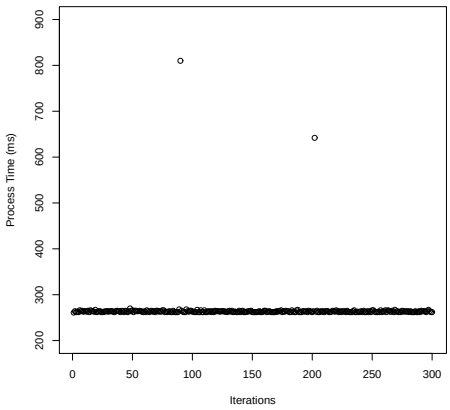
(a) PUT64's PTs



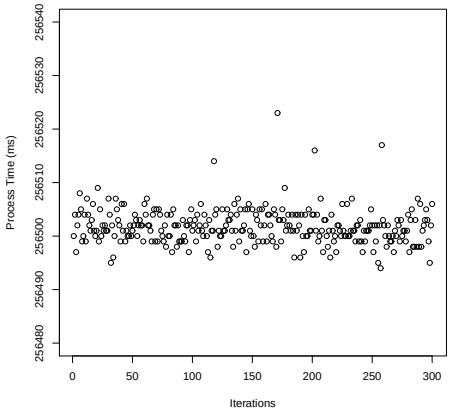
(b) PUT64's daemon PTs



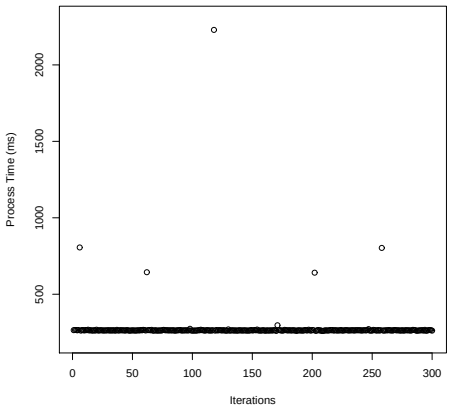
(c) PUT128's PTs



(d) PUT128's daemon PTs

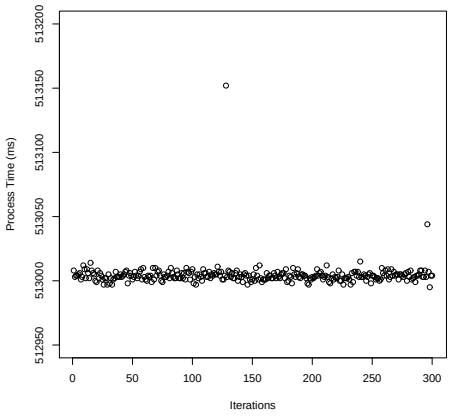


(e) PUT256's PTs

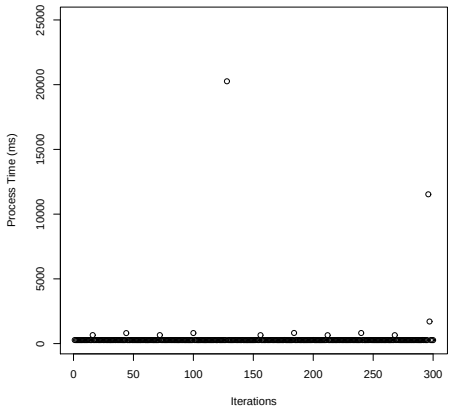


(f) PUT256's daemon PTs

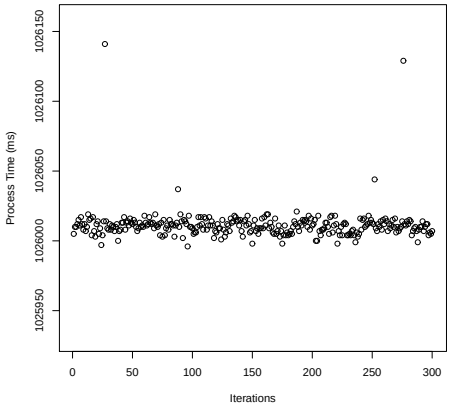
Figure 23: Program times between PUT64∼PUT256 vs. Daemon processes



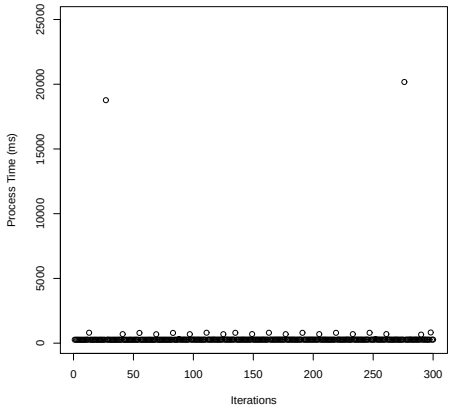
(a) PUT512's PTs



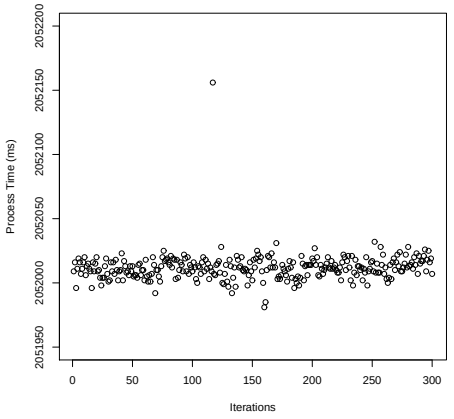
(b) PUT512's daemon PTs



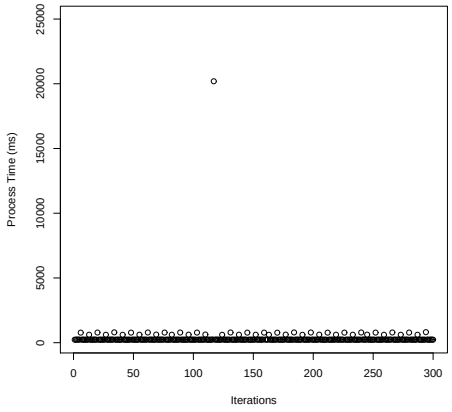
(c) PUT1024's PTs



(d) PUT1024's daemon PTs

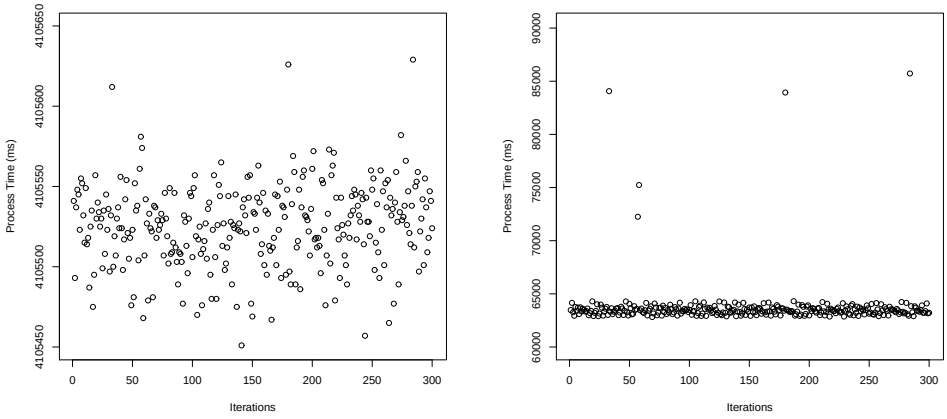


(e) PUT2048's PTs



(f) PUT2048's daemon PTs

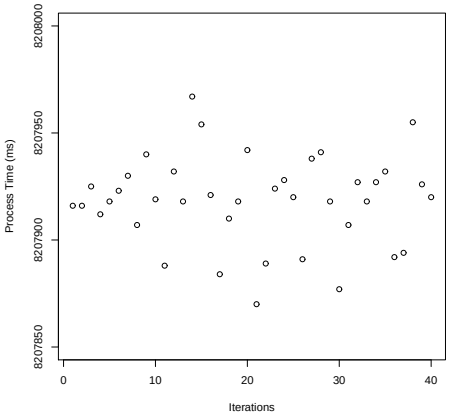
Figure 24: Program times between PUT512∼PUT2048 vs. Daemon processes



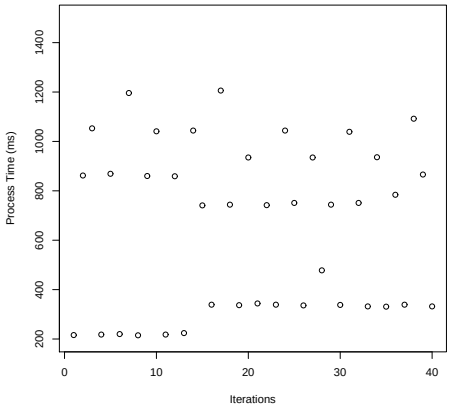
(a) PUT4096's PTs

(b) PUT4096's daemon PTs

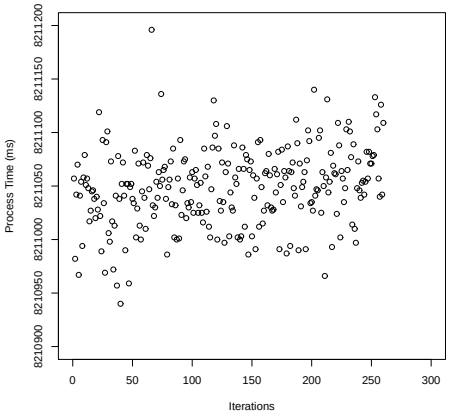
Figure 25: Program times between PUT4096 vs. Daemon processes



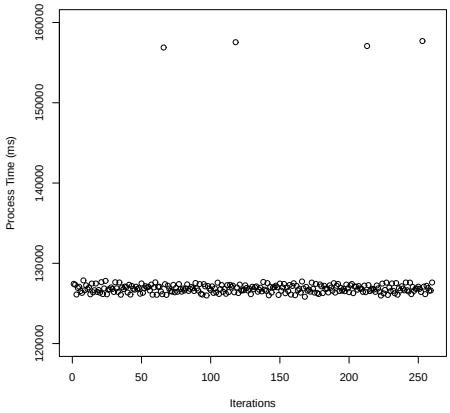
(a) PUT8192's PTs in Apr 2015



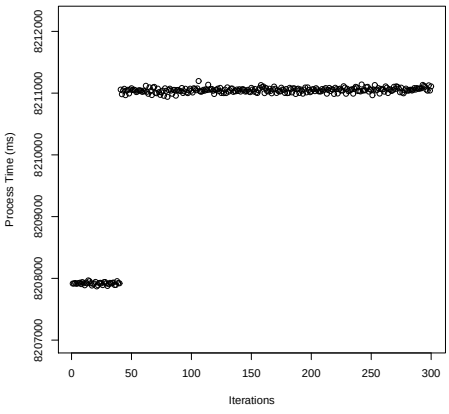
(b) PUT8192's daemon PTs in Apr 2015



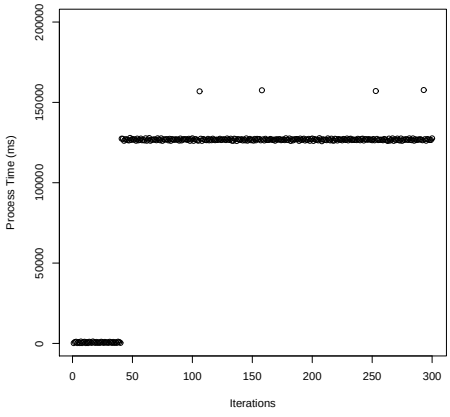
(c) PUT8192's PTs in Nov 2015



(d) PUT8192's daemon PTs in Nov 2015

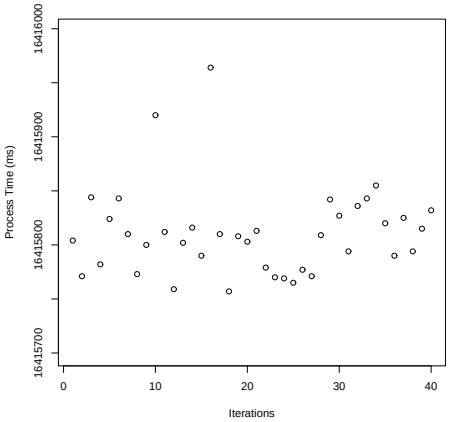


(e) Combined PUT8192's PTs

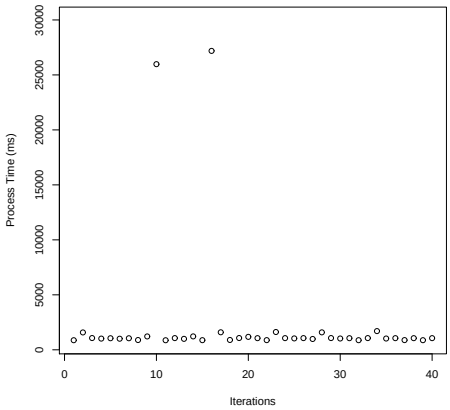


(f) Combined PUT8192's daemon PTs

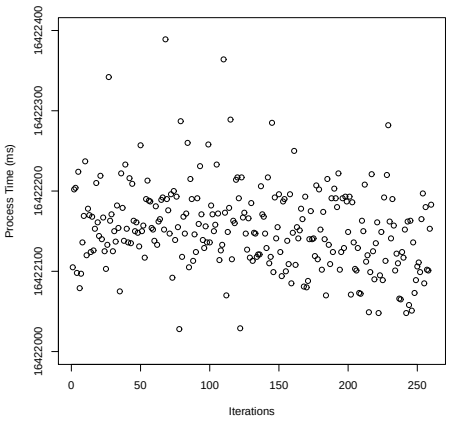
Figure 26: Program times between PUT8192 vs. Daemon processes



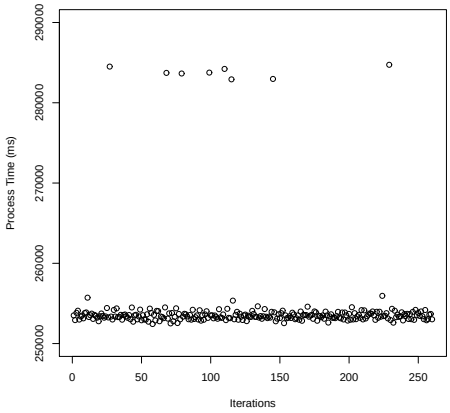
(a) PUT16384's PTs in Apr 2015



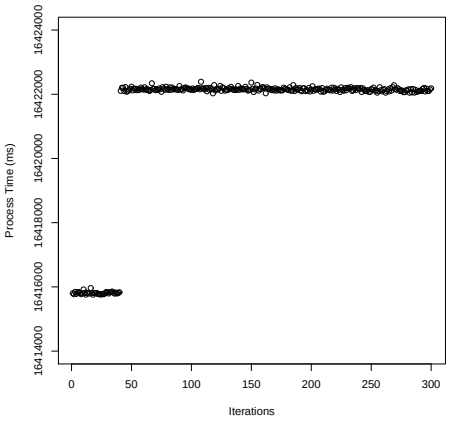
(b) PUT16384's daemon PTs in Apr 2015



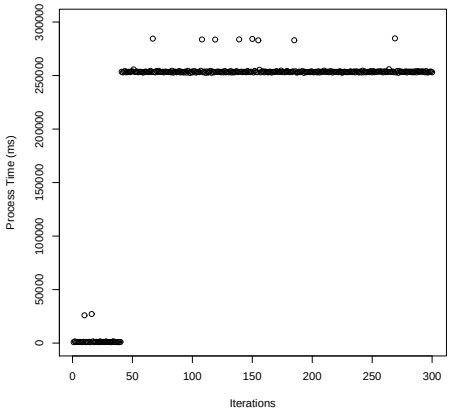
(c) PUT16384's PTs in Nov 2015



(d) PUT16384's daemon PTs in Nov 2015



(e) Combined PUT16384's PTs



(f) Combined PUT16384's daemon PTs

Figure 27: Program times between PUT16384 vs. Daemon processes

8 Appendix

8.1 Breakdown on Program Times of Daemon Processes

PUT256	Program Time
`incr_work`	256,514 msecs (at the 118th iteration)
Daemon Processes	Program Time
`java`	2 msecs
`md0_raid1`	4 msecs
`jbd2/md0-8`	1 msec
`flush-9:0`	10 msecs
`proc_monitor`	262 msecs
`rhnsd`	6 msecs
`rhn_check`	1,944 msecs
Total	2,229 msecs

Table 8: The daemon processes captured at the worst PT of PUT256

PUT256	Program Time
`incr_work`	513,152 msecs (at the 128th iteration)
Daemon Processes	Program Time
`java`	2 msecs
`md0_raid1`	51 msecs
`jbd2/md0-8`	27 msecs
`flush-9:0`	86 msecs
`proc_monitor`	270 msecs
`rhnsd`	6 msecs
`rhn_check`	19,820 msecs
Total	20,262 msecs

Table 9: The daemon processes captured at the worst PT of PUT512

PUT4096	Program Time
`incr_work`	4,105,629 msecs (at the 284th iteration)
Daemon Processes	Program Time
`events/0`	1 msec
`kblockd/0`	1 msec
`kslowd000`	31,710 msecs
`kslowd001`	31,782 msecs
`md0_raid1`	82 msecs
`jbd2/md0-8`	21 msecs
`flush-9:0`	79 msecs
`proc_monitor`	206 msecs
`rhnsd`	3 msecs
`ntpd`	1 msec
`java`	2 msecs
`rhn_check`	21,840 msecs
Total	85,728 msecs

Table 10: The daemon processes captured at the worst PT of PUT4096

PUT8192	Program Time
incr_work	8,207,884 msecs (at the 244th iteration)
Daemon Processes	Program Time
kblockd/0	3 msecs
kslowd000	31,710 msecs
kslowd001	31,782 msecs
md0_raid1	12 msecs
jbd2/md0-8	2 msecs
proc_monitor	204 msecs
rhnsd	6 msecs
java	1 msec
rhsmcertd-worke	114 msecs
rhsmcertd-worke	114 msecs
rhn_check	708 msecs
Total	64,656 msecs

Table 11: The daemon processes captured at the worst PT of PUT8192

Daemon Processes	Descriptions
kslowd000 (kslowd001)	A kernel threads for performing things that take a relatively long time. "Typically, when processing something, these items will spend a lot of time, blocking a thread on I/O, thus making that thread unavailable for doing other work." (http://www.mjmwired.net/kernel/Documentation/slow-work.txt)
rhn_check	An external program for check for updates, run by rhnsd
rhnsd	"A background daemon process that periodically polls the Red Hat Network to see if there are any queued actions available. Typically started from the initialization (init) scripts in /etc/init.d/rhnsd when its time to poll the Red Hat Network server for available updates and actions. The default interval is every 240 minutes. The minimum polling interval is 60 minutes. Any network activity is done via the rhn_check utility." (http://linuxcommand.org/man_pages/rhnsd8.html)

Table 12: Descriptions of some daemon processes