

**Measuring Differences in SAS Computation Time
when Running a Single Thread Program One at a
Time vs. Two at a Time**

James D. Abbey
May 25, 2004

1 - Introduction

1.1 – Purpose of Experiment

This experiment seeks differences in computational speed between the AMD Opteron and Intel Xeon server processors under nearly identical conditions when running a completely self contained and highly computationally intense probability simulation. The computational speed is measured in seconds. The experiment tests the speed of the two architectures when running either one single threaded program across two CPUs or two identical single threaded programs across two CPUs. The Xeon is faster when running a single threaded program. Yet, the Opteron is faster when running two single threaded programs simultaneously.

1.2 – Audience for the Results

The target audience for this research includes SAS users and server administrators. Also, this research can be used by MIS and technical people in other fields. However, some familiarity with SAS and mathematical concepts is required for full understanding. In addition, further research regarding much higher levels of programs (4, 8 and 16 simultaneous programs) is underway.

1.3 – Testbeds

Operating System: Windows 2000 sp 4. Hyperthreading was disabled in the Xeon because Windows 2000 does not support this feature.

Intel Xeon Hardware:

Processors: Two Intel Xeon 2.4Ghz 533Mhz Quad Pumped Bus Socket 604

Memory: 2GB Buffalo Technology PC3200 Registered ECC DDR (Samsung Chips)

Motherboard: Supermicro X5DP8-G2 Intel E7501 Dual Xeon

RAID Card: Adaptec 2010S 48MB Onboard Cache

Hard Drives: Fujitsu MAS 15,000rpm 36.4GB (2 in JBOD)

Power Supply: Antec 550Watt 24-pin Power Supplies

Chassis: Lian-Li Cases PC-73SL

AMD Opteron Hardware:

Processors: Two AMD Opteron 242 (1.6Ghz) 800Mhz Hyper Transport Socket 940

Memory: 2 GB Buffalo Technology PC3200 Registered ECC DDR (Samsung Chips)

Motherboard: Tyan S2882UG3NR K8SPro AMD 8131/8111 Chipsets

RAID Card: Adaptec 2015S 48MB Onboard Cache

Hard Drives: Fujitsu MAS 15,000rpm 36.4GB (2 in JBOD)

Power Supply: Antec 550Watt 24-pin Power Supplies

Chassis: Lian-Li Cases PC-73SL

The choice of hardware is based entirely upon comparable cost. Both servers' totaled around \$2600 each in April, 2004. The Xeon was slightly more expensive due solely to a

\$15 per processor higher cost. Further, all other components are identical or nearly so except the system motherboards. Finally, though the servers cost \$2600 to *build by hand*, comparable servers from IBM and Dell cost nearly twice as much for nearly identical, if not slightly lower quality, components.

2 – Experiment Details

2.1 – Experimental Program

The program used for all runs uses a $\binom{q^n}{M}$ probability formula to select a usable number of designs. A vector based formula tests each design. All runs used the levels $\binom{2^5}{8}$ giving 10,518,300 tested samples designs for each run.

2.2 – Experimental Design

This experiment used the Split Plot/Repeated Measures (SP/RM) experimental design. A breakdown of the factors follows:

- *Response*: Time to complete the program in seconds
- *Wholeplot Factor*: CPU Architecture [Dual Opteron or Dual Xeon]
- *Subplot Factor*: Number of single threaded programs running [1 or 2]

2.3 – Experimental Details and Description

Experimental Material: The testbeds and program information can be found above in sections 1.3 and 2.1, respectively.

Experimental Control: The environment for this experiment is a stable temperature dehumidified basement computer room. Both systems sit in identical Lian-Li PC-73SL cases. Both cases use the same style of Antec 550 Watt power supply. Nearly all components are identical across systems to keep external variables from altering the results. At the time this experiment was conducted, the Fujitsu MAS 36.5GB 15,000rpm drives were the fastest server drives on the market according to storagereview.com.¹

Experimental Replication: A total of 24 runs were completed. Thus, each treatment level had 6 replicates. Each factor level had 12 replicates. This gives a treatment level alpha = 0.05, beta = 0.10 and significant difference of 2.5 deviations. Further, the factor levels had alpha = 0.05, beta = 0.05 and a significant difference of 1.6 deviations.

Experimental Randomization: To randomize in a SP/RM design, you first randomize the wholeplot factor as a completely randomized variable. The order of runs was randomized

¹ Storagereview.com: http://www.storagereview.com/articles/200304/20030429MAS3735_1.html

by entering the runs in column one and then creating and then sorting by random digits in column two. The order of the wholeplot factors appears below:

CPU Type	Whole Plot Rnd Dig
Opteron	0.03233392
Xeon	0.04423746
Xeon	0.10705824
Xeon	0.14171816
Opteron	0.18540456
Opteron	0.29004017
Opteron	0.35597217
Xeon	0.4277189
Xeon	0.45275263
Xeon	0.57361947
Opteron	0.76018372
Opteron	0.80494697

(2.3.1)

In addition, the subplot factors are randomized within the wholeplot factor. Because the subplot factor has only two levels, each order was determined by flipping a coin. The order of the number of programs was as follows:

CPU Type	Run One	Run Two
Opteron	2	1
Xeon	1	2
Xeon	1	2
Xeon	2	1
Opteron	2	1
Opteron	1	2
Opteron	1	2
Xeon	2	1
Xeon	1	2
Xeon	1	2
Opteron	2	1
Opteron	1	2

(2.3.2)

Experimental Process: Each run entailed opening a fresh copy of SAS by double clicking the desktop icon. Next, any changes to the output file and log file names were made, and the program was saved. The program was then run and timed. Times were recorded as reported by the Windows 2000 task manager. In this experiment, real time and the CPU time were identical. After all measurement were complete, SAS was shutdown and the process repeated.

3 – Data Analysis and Discussion

3.1 – Treatment Means, Plots and t-tests

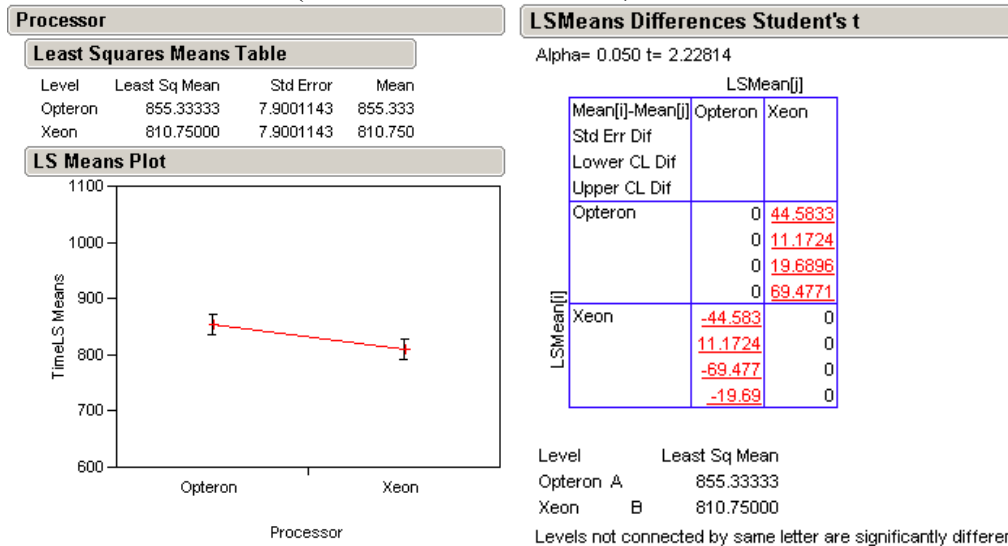
Treatment and Factor Level Means: The following are the treatment factor level means, standard errors and LSMeans plots for CPU Architecture and Number of Programs.

- Overall F-Stats and p-values:

Source	SS	MS Num	DF Num	F Ratio	Prob > F
Processor	11926	11926	1	15.9239	0.0026
Run Number[Processor]&Random	7489.42	748.942	10	2.1471	0.1220
Number of Programs	251126	251126	1	719.9542	<.0001
Processor*Number of Programs	140913	140913	1	403.9851	<.0001

(3.1.1)

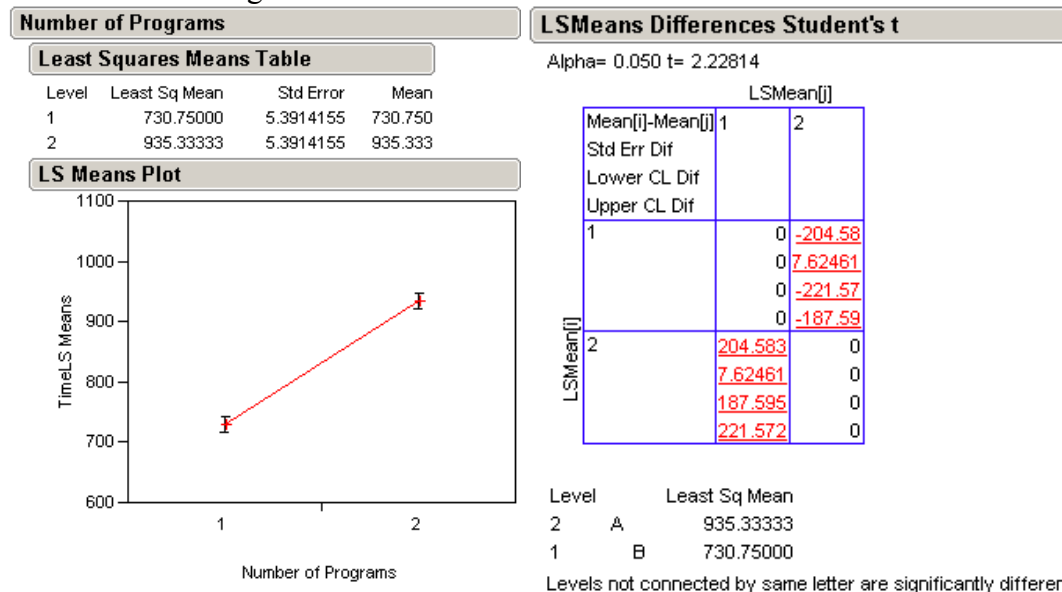
- CPU Architecture (lower numbers are better):



(3.1.2)

The results show that the Xeon is the faster processor on average, coming in at approximately 811 seconds (13.5 minutes). The Opteron was only a bit slower at an average of 855 seconds (14.25 minutes). The levels were statistically significantly different at a 95% (alpha = 0.05) confidence level using a t-test with Ho: CPU architecture factor level means equal.

- Number of Programs:



(3.1.3)

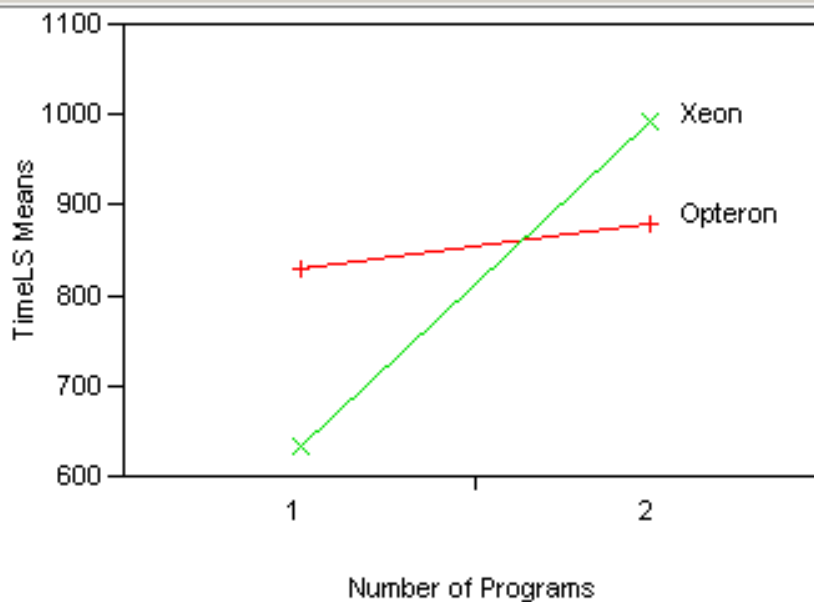
The results display that running two programs does indeed take longer than running one. However, the *average time per program diminished significantly* when running two programs simultaneously averaging only $935/2 = 467.5$ seconds or **7.8 minutes per output result vs. 730.75 or 12.175 minutes when running only one program at a time**. Further, note that the factor level means showed a statistically significant difference at the alpha = 0.05 or 95% confidence level under a t-test with Ho: number of programs factor means equal.

- Treatment means and Interaction Plot:

Least Squares Means Table

Level	Least Sq Mean	Std Error
Opteron,1	829.66667	7.6246129
Opteron,2	881.00000	7.6246129
Xeon,1	631.83333	7.6246129
Xeon,2	989.66667	7.6246129

LS Means Plot

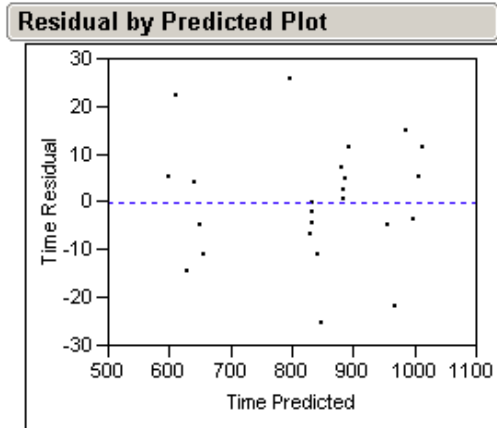


(3.1.4)

The interaction between processor architecture (Xeon or Opteron) and number of programs (one or two) appears unmistakable. The levels show a distinct “X” pattern. This X pattern indicates that the Opteron is the slower chip when running one program, but when two identical number crunching programs were running simultaneously, the Opteron was faster. Further, the Opteron shows very little increase in total time due to running two programs as opposed to one. The Opteron’s averages moved only from 830 seconds to 881 seconds when running one and two programs, respectively. The Xeon did not handle the increased load as well with average time moving from 632 seconds with one program to 990 seconds with two programs. Tests are underway to determine if this pattern holds for 4, 8 and even 16 simultaneous programs.

3.2 – Residuals Analysis

Residual by Predicted Plot: One of the six ANOVA assumptions is that the variance of any one predicted value should be no more than three times larger than the smallest variance of a predicted value. The following plot allows graphical analysis of the assumption:

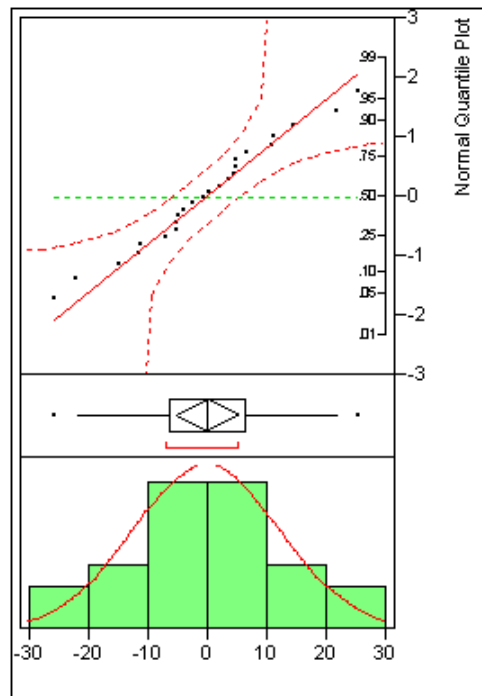


(3.2.1)

The condition of equal variance may have been violated, but due to the strong F-statistics and p-values reported above, this violation does not appear to harm the results of the experiment.

Normality of the Residual Distribution: Another of the ANOVA assumptions is that the residuals are normally distributed. To check for this normality, the normal quantile plot and histogram of the residuals appears to the right:

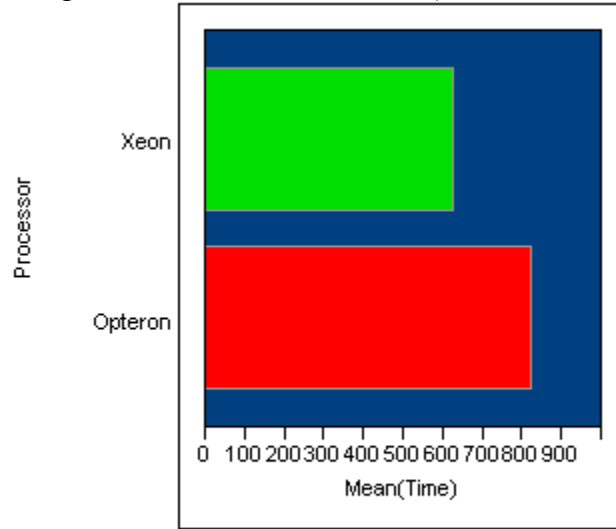
The normality assumption appear met. Though two outliers are reported, the overall pattern is quite normal. No point seems particularly unusual or out of place.



(3.2.2)

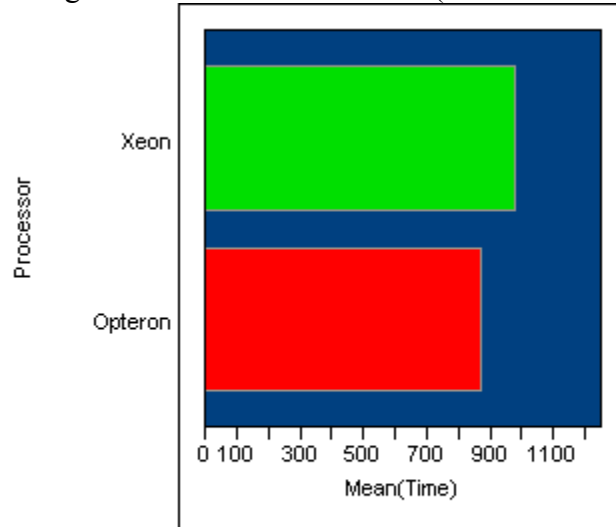
3.3 – Summary of Results in Graphical Format

- One Program Mean Execution Time (lower is better):



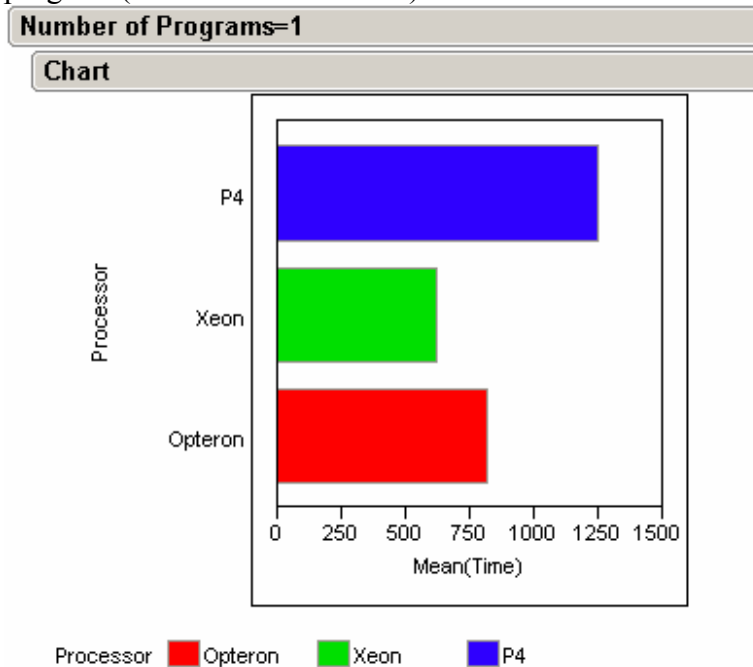
Processor ■ Opteron ■ Xeon

- Two Program Mean Execution Time (lower is better):



Processor ■ Opteron ■ Xeon

- Comparison of a Pentium 4, Opteron and Xeon system mean time to complete one program (lower times are better):



The older Pentium 4 single processor system turned in an average time of 1260 seconds or exactly 21:00 minutes over six total runs. Additionally, the Pentium 4 was remarkably stable with a standard deviation of only 0.6324. Yet, the Xeon at 632 seconds and the Opteron at 830 seconds clearly dominated the older Pentium 4 1.6Ghz system. See the Pentium 4 specifications below. Hence, the Xeon approximately halves the amount of time the single number crunching program takes to run when compared to the P4 1.6GHz. Further, the Opteron saves approximately 430 seconds or 7.16 minutes running one number crunching program when compared to the P4 1.6Ghz.

Pentium 4 Hardware:

Processor: Intel Pentium 4 1.6Ghz socket 478

Memory: 1GB Crucial PC-2100

Motherboard: ASUS P4B533-VM

Hard Drive: Maxtor 40 GB ATA-66 7200rpm

Power Supply: Turbolink 420 Watt Power Supply

Chassis: Chenming Full Tower (Antec SX1040 Server Tower Clone)

4 – Conclusion

If you are running one program at a time, the Intel Xeon is likely the better single program number cruncher. Because this program was not multi-threaded, you do not need two Xeon processors when running only one program. However, when you run the two iterations of same number crunching program simultaneously, the Opteron appears to

be the better choice when using dual processors. Hence, for the individual user who merely wants single program runs, the Xeon appears to be the better choice. However, the server administrator or multiple program user may want to consider using the Opteron.

When compared to the older Pentium 4 1.6GHz computer, both servers turned in significantly faster times. Most users of intense single threaded single run programs would be well advised to upgrade to the 2.4GHz Xeon, which was twice as fast as the 1.6GHz P4. As before, users of two simultaneous single threaded programs may wish to consider the dual Opteron.