Estimation of Ready Queue Processing Time using Transformed Factor-Type (T-F-T) Estimator in Multiprocessor Environment

Diwakar Shukla
Deptt. of Mathematics and Statistics, H.S. Gour Central University, Sagar, M.P, India

Anjali Jain
Deptt. of Computer Sc. and Applications
H.S. Gour Central University
Sagar, M.P, India

Kapil Verma
Deptt. of Computer Science
M. P. Bhoj (Open) University
Bhopal, M.P, India

ABSTRACT

The ready queue processing time estimation problem appears when many processes remain in the ready queue just before the occurrence of sudden failure of system. The system administrator has to decide immediately, how much further time is required to process the remaining jobs in the ready queue, before shutting down the entire system as precautionary measure, so that while restart, it may remain in the safe state. In lottery scheduling, this prediction is possible with the help of sampling techniques. Factor- Type estimation method, existing in literature of sampling, was used by many authors to predict the time required provided the highly correlated sources of auxiliary information are available. This paper suggests two new estimation methods to predict the remaining total processing time required to process completely the ready queue provided sources of auxiliary information are negatively correlated. Under this approximation, the bias and m.s.e of the proposed estimators have been obtained using the set up of random sampling applicable to lottery scheduling. Performance of both estimation methods are compared in terms of mean squared error. The confidence intervals are calculated for comparing the efficiency of the estimate. One proposed estimator found better over other. .

Keywords

Lottery Scheduling, Transformed Factor-Type (T-F-T) Estimator, Mean Squared Error (M.S.E), Variance, Confidence Intervals

1. INTRODUCTION

Suppose there are k processors in a multiprocessor and multi-user environment and a large number of processes, say *N*, are in waiting queue. The CPU scheduler adopts lottery scheduling procedure to choose randomly any *n* processes from the waiting queue $\left(n < N\right)$ and allocates to *k* processors $\left(n \approx k\right)$ in sequential manner. The Lottery scheduling is different from basic scheduling algorithm, where each process is allocated a number of lottery tickets determining the possibility of process when to use the CPU. At each schedule point, a lottery is held and the process in the ready queue with the winning ticket gets the CPU utilization. Unlike priority scheduling, every job has equal chance of being represented to the processors [1] [2].

A technical problem appears when N is very large and because of the congestion, many processes have to wait until they called in random manner. If all of sudden the system collapses due to failure of power supply, maintenance problem, technical faults or due to any other unavoidable reason, the system manager has to look for backup management. His problem at this juncture is to know how much additional time require for finishing the remaining processes so that after restart the system remains in safe state. Such predictions are uncertain and require probability mechanism to resolve. Processor queues are matrices and Microsoft Windows (NT, 2000, XP) maintains its registry.

To predict system behavior and CPU queue length, Microsoft NT, Windows 2000 developed a queuing theory model and derived fair share scheduling which guarantees application performance by explicitly allocating share of system resources among competing workloads [3]. Observed values (auxiliary information) often contradict expectations about how queue length should correlate to the system load (e.g. CPU utilization or burst time, transaction response time, process priority etc.) [4,5]. In literature it is common and well known idea that often more input information provides better prediction subject to condition when information are related whether positively or negatively [6, 7]. In order to obtain ready queue length time estimate, the authors in [8] proposed usual lottery scheduling (ULS) procedures in multiprocessor environment where variable of main interest (i.e. processing time) and auxiliary information like size of the process was positively correlated.

In [9], the authors describes systematic lottery SL-Scheduling scheme, to improve upon the prediction of ready queue processing time.

Similar problem was considered in [10] in lottery scheduling setup when processes are grouped according to specific criteria in different queues and ready queue processing time estimates were obtained. However, there are some procedures of deciding priorities among jobs maintaining the fairness in selection methods [11].

The size based priority scheme [12] PPS-LS, for the ready queue time length prediction has shown that it is better than usual lottery scheduling scheme in terms of confidence intervals. In [13], two generalized methods of estimation to compute the predicted time intervals were suggested and compared in terms of variances, confidence intervals. The authors have proposed efficient factor-type (E-F-T) estimator in order to predict efficiently time intervals in setup of lottery scheduling and considered the earlier similar problem. The proposed methods of estimation in [8-10] and [12-13] have shown improvement over one another. However, they did not consider the case of negative correlation between variables. Based on this thought, this paper introduces the application of Transformed Factor-Type (T-F-T) estimator and computes predicted time interval i.e. time required for processing the remaining jobs, when auxiliary sources of information are negatively correlated to the main variable. In content the two estimation procedures are compared in terms of their performances with the help of [14, 15].

2. MODIFIED USUAL LOTTERY SCHEDULING (AS PER [4])

Step1. When a process enters into ready queue, it is allotted a token numbers (in specified range).

Step2. Each processor $Q_1, Q_2, Q_3 .... Q_k .... Q_K$ generates unique and uncommon random number in similar specified range as stated in Step 1.

Step3. Matching of token numbers of process and random number of processor takes place by the scheduler. If both numbers are same for a process in ready queue, it is assigned to that processor.

Step4. Repeat step 1, step 2 and step 3 in order to select another process in random manner until ready queue is vacated.

3. NOTATIONS AND PROCESSOR STRUCTURE

N: Total processes currently in the ready queue.

n: Randomly selected processes for processing before breakdown.

f **:** Sampling fraction i.e. (n / N)

X_i: Value of the auxiliary information like priority of ith process.

Y_i: CPU burst time of each ith process $(i = 1,2,3.....N)$.

$\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i$: The mean of the CPU burst time of N processes in the ready queue.

$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$: The mean of priorities of N processes in the ready queue.

$C_Y^2 = \frac{S_Y^2}{\bar{Y}^2}$: Coefficient of variation for process population parameter Y.

$C_X^2 = \frac{S_X^2}{\bar{X}}$: Coefficient of variation for process population parameter X.

$C_{XY} = \frac{S_{XY}}{\bar{X}\bar{Y}}$: Coefficient of variation due to X and Y.

There are $Q_1, Q_2, Q_3 ... Q_k .... Q_K$ K processors who receive in take from the ready queue containing $P_1, P_2, P_3 .... P_n .... P_N$ processes $(n < N)$. When a process is blocked or suspended or interrupted, it is placed back to the respective queue [16]. The Figure 1 shows formation of different possible samples, through lottery scheduling of n processes from the ready queue, having pool of N processes, where k processors will get any sample of processes of varying sizes. Processes are processed completely or partially, before breakdown and processors generate time consumed in processing as denoted y_j (time by j^{th} processor) where $(j = 1,2,3,..., k,....K)$.

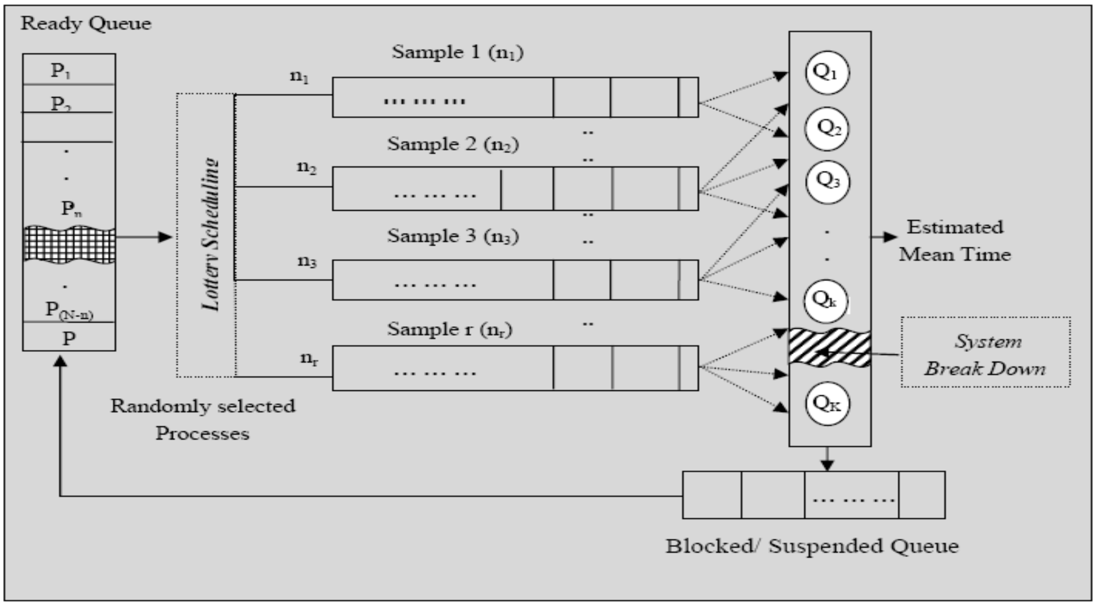
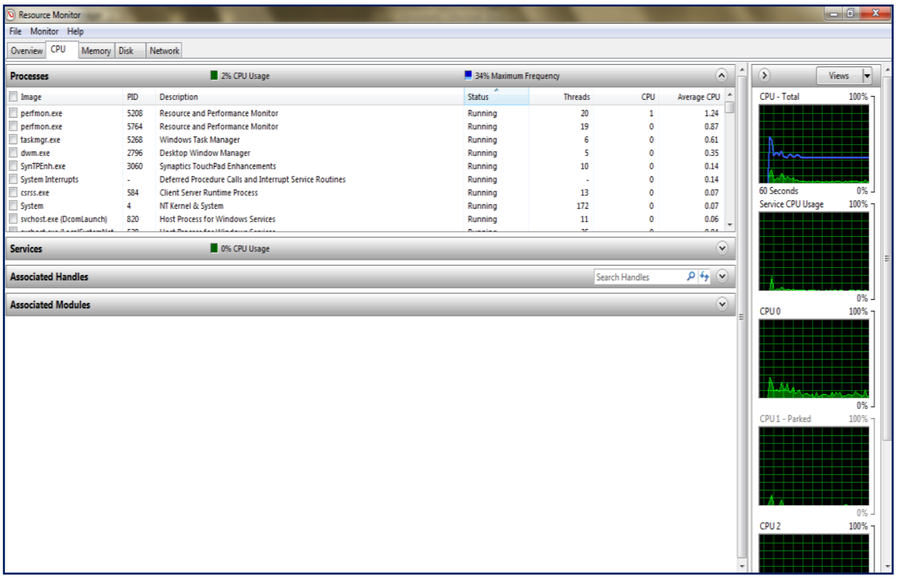


Figure 1. Lottery scheduling of processes to *k* processors with estimation of expected time after breakdown.

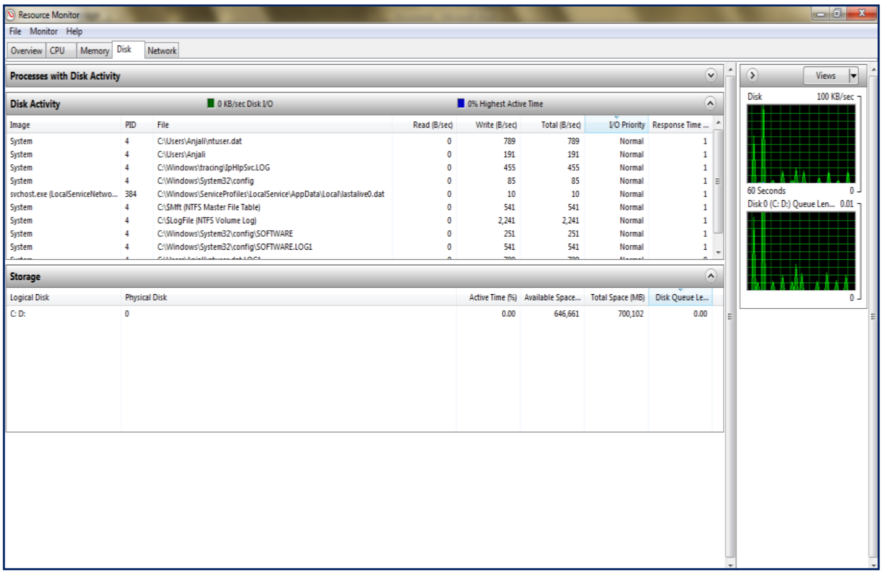
Other available information of processes is x_i as a source of auxiliary variable. Figure 2 and Figure 3, shows possible sources of auxiliary information of processes currently running in the system.



Processes ■ 2% CPU Usage ■ 34% Maximum Frequency

Image	PID	Description	Status	Threads	CPU	Average CPU
□ perfmon.exe	5208	Resource and Performance Monitor	Running	20	1	1.24
□ perfmon.exe	5764	Resource and Performance Monitor	Running	19	0	0.87
□ taskmgr.exe	5268	Windows Task Manager	Running	6	0	0.61
□ dwm.exe	2796	Desktop Window Manager	Running	5	0	0.35
□ SynTPEnh.exe	3060	Synaptics TouchPad Enhancements	Running	10	0	0.14
□ System Interrupts	-	Deferred Procedure Calls and Interrupt Service Routines	Running	-	0	0.14
□ csrss.exe	584	Client Server Runtime Process	Running	13	0	0.07
□ System	4	NT Kernel & System	Running	172	0	0.07
□ svchost.exe (DcomLaunch)	820	Host Process for Windows Services	Running	11	0	0.06

Figure 2. Active processes in the system.


Resource Monitor
File Monitor Help
Overview CPU Memory Disk Network
Processes with Disk Activity
Views
Disk 100 KB/sec
60 Seconds 0
Disk 0 (C: D:) Queue Len... 0.01
0

Disk Activity ■ 0 KB/sec Disk I/O ■ 0% Highest Active Time

Image	PID	File	Read (B/sec)	Write (B/sec)	Total (B/sec)	I/O Priority	Response Time ...
System	4	C:\Users\Anjali\ntuser.dat	0	789	789	Normal	1
System	4	C:\Users\Anjali	0	191	191	Normal	1
System	4	C:\Windows\tracing\IpHlpSvc.LOG	0	455	455	Normal	1
System	4	C:\Windows\System32\config	0	85	85	Normal	1
svchost.exe (LocalServiceNetwo...	384	C:\Windows\ServiceProfiles\LocalService\AppData\Local\lastalive0.dat	0	10	10	Normal	1
System	4	C:\$Mft (NTFS Master File Table)	0	541	541	Normal	1
System	4	C:\$LogFile (NTFS Volume Log)	0	2,241	2,241	Normal	1
System	4	C:\Windows\System32\config\SOFTWARE	0	251	251	Normal	1
System	4	C:\Windows\System32\config\SOFTWARE.LOG1	0	541	541	Normal	1

Storage

Logical Disk	Physical Disk	Active Time (%)	Available Space...	Total Space (MB)	Disk Queue Le...
C: D:	0	0.00	646,661	700,102	0.00

Figure 3. Processes with Disk Activity and Priorities.

4. ESTIMATION OF READY QUEUE PROCESSING TIME (AS PER [17])

We know Ratio estimator:

$$B\left(\bar{y}_r\right) = \bar{Y}\left(V_{02} - V_{11}\right) \qquad ...(1)$$

$$M\left(\bar{y}_r\right) = \bar{Y}^2\left(V_{20} - 2V_{11} + V_{02}\right) \qquad ...(2)$$

..(2)

Consider following notations;

$$z_i = d\bar{X} - x_i, \left(i = 1, 2, ..., N\right) where \quad z_i > 0$$

$$\bar{Z} = (d-1)\bar{X}$$

$$A = (d-1)\ (d-2);$$

$$B = (d-1)\ (d-4);$$

$$C = (d-2)\ (d-3)\ (d-4);$$

where d is a constant $(0 < d < \infty)$. Transformed Factor-Type *(T-F-T)* estimator in [18] for $\bar{Y}$, is

$$T_{TFT} = \bar{y}\left[\frac{(A+C)\bar{Z} + fB\bar{z}}{(A+fB)\bar{Z} + C\bar{z}}\right] \qquad ...(3)$$

$$= \bar{y}\left[\frac{(A+C)(d-1)\bar{X} + fB(d\bar{X} - \bar{x})}{(A+fB)(d-1)\bar{X} + C\left(d\bar{X} - \bar{x}\right)}\right] \qquad ...(4)$$

Consider the large sample approximation $\bar{y} = \bar{Y}(1+e_0), \bar{x} = \bar{X}(1+e_1)$, so that $E(e_0) = E(e_1) = 0,$

$$E\left(e_0^i e_1^j\right) = V_{ij},\ i,\ j\ =\ 0, 1, 2.$$

In above expressions

$$V_{ij} = E\left(\bar{y} - \bar{Y}\right)^i \left(\bar{x} - \bar{X}\right)^j / \bar{Y}^{\,i} \bar{X}^{\,j} ; \bar{Y}, \bar{X} \neq 0 \ldots(5)$$

$$V = \left(V_{11} / V_{02}\right) = \rho C_Y / C_X$$

Using notation Bias () for bias and M.S.E () for mean squared error, since the term

$$\left| \frac{Ce_1}{(A + fB + C)(d - 1)} \right| < 1$$

for our choice of A, B and C; We get the following expressions

$$Bias\left(T_{TFT}\right) = \bar{Y}\left[Q\left\{\phi_1 V_{02} + V_{11}\right\}\right]$$

$$M.S.E\left(T_{TFT}\right) = \bar{Y}^2\left[V_{20} + Q^2 V_{02} + 2QV_{11}\right]$$

where $Q = \phi_1 - \phi_2$

$$\phi_1 = \frac{C}{\left[(A + fB + C)(d - 1)\right]}$$

$$\phi_2 = \frac{fB}{\left[(A + fB + C)(d - 1)\right]}$$

4.1. Suggested Ready Queue Processing Time Estimators With Numerical Illustration

Using [14] and [15], we suggest the two estimators as below;

At $d = 5$, we choose one method from class T_{TFT} marked as T_A,

$$T_A = \bar{y}\left[\frac{(72 + 20f)\bar{X} - 4f\bar{x}}{90\bar{X} + (20f - 1)\bar{x}}\right] \ldots(6)$$

$$Bias\left(T_A\right) = \bar{Y}\left[\frac{3 - 2f}{4(9 + 2f)}\left\{\frac{3}{4(9 + 2f)}V_{02} + V_{11}\right\}\right] \ldots(7)$$

$$M.S.E\left(T_A\right) = \bar{Y}^2\left[\begin{array}{l} V_{20} + \left(\dfrac{3 - 2f}{4(9 + 2f)}\right)^2 V_{02} \\ + 2\left(\dfrac{3 - 2f}{4(9 + 2f)}\right)V_{11} \end{array}\right] \ldots(8)$$

Similarly, at $d = 6$, we choose another method in class T_{TFT} marked as T_B

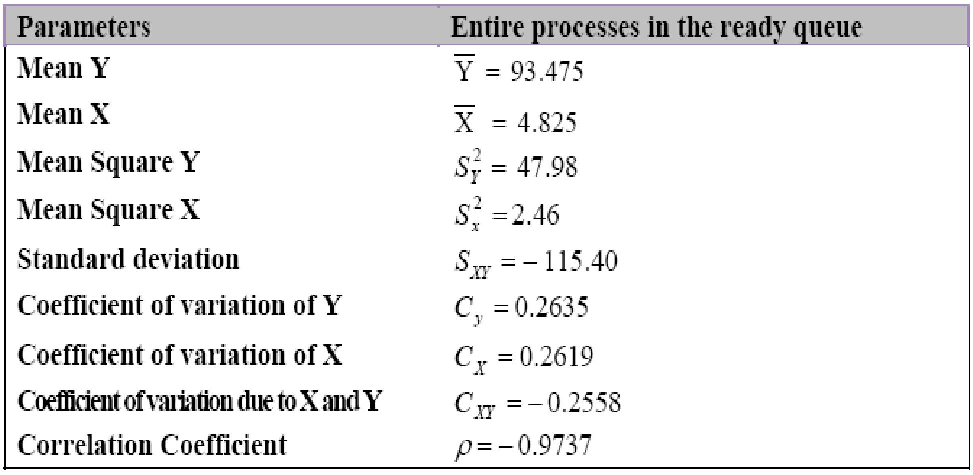
$$T_B = \bar{y}\left[\frac{(110 + 30f)\bar{X} - 5f\bar{x}}{(122 + 25f)\bar{X} - 12\bar{x}}\right] \ldots(9)$$

$$Bias\left(T_B\right) = \bar{Y}\left[\frac{12 - 5f}{5(22 + 5f)}\left\{\frac{12}{5(22 + 5f)}V_{02} + V_{11}\right\}\right] \ldots(10)$$

$$M.S.E\left(T_B\right) = \bar{Y}^2\left[\begin{array}{l} V_{20} + \left(\dfrac{12 - 5f}{5(22 + 5f)}\right)^2 V_{02} \\ + 2\left(\dfrac{12 - 5f}{5(22 + 5f)}\right)V_{11} \end{array}\right] \ldots(11)$$

We assume that (N = 40) processes in the ready queue at a time in the system and *n* processes are currently running in the system at a time, whose snap shot of the CPU burst/usage time in seconds as Y and priority values as X is mentioned against them, displayed in Appendix A. The process population parameters are shown in Table 1.

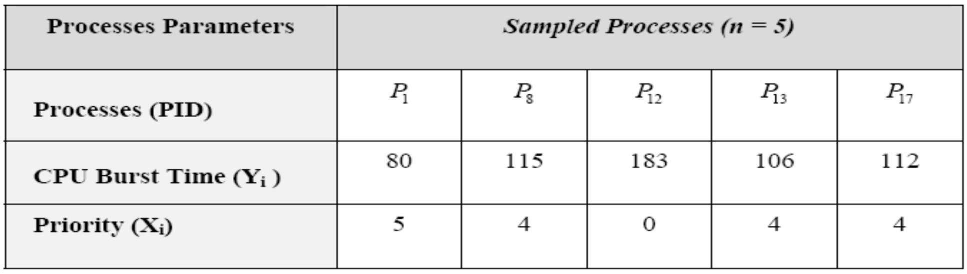
Table 1. Processes population parameters for N = 40 (from Appendix A).



Parameters	Entire processes in the ready queue
Mean Y	$\bar{Y} = 93.475$
Mean X	$\bar{X} = 4.825$
Mean Square Y	$S_Y^2 = 47.98$
Mean Square X	$S_x^2 = 2.46$
Standard deviation	$S_{XY} = -115.40$
Coefficient of variation of Y	$C_y = 0.2635$
Coefficient of variation of X	$C_X = 0.2619$
Coefficient of variation due to X and Y	$C_{XY} = -0.2558$
Correlation Coefficient	$\rho = -0.9737$

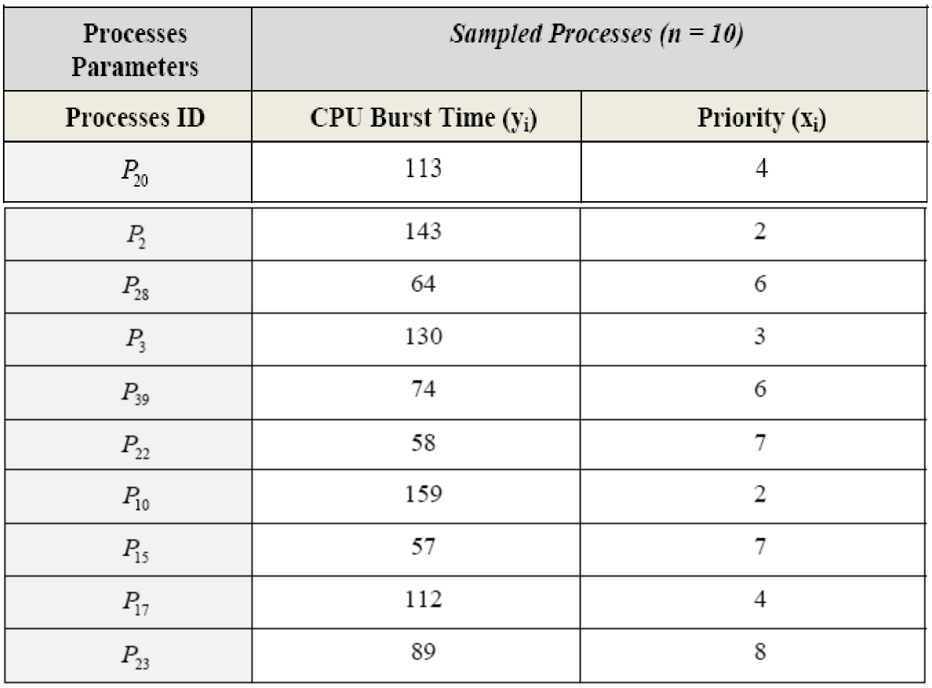
Suppose out of N processes, n are processed ($n < N$) and remaining ($N - n$) are still in the system unprocessed, when sudden disaster/collapse occurs. As an example Table 2, Table 3 and Table 4 shows samples of n ($n = 5, 10, 15$) processed processes before breakdown.

Table 2. Description of first sample n = 5.



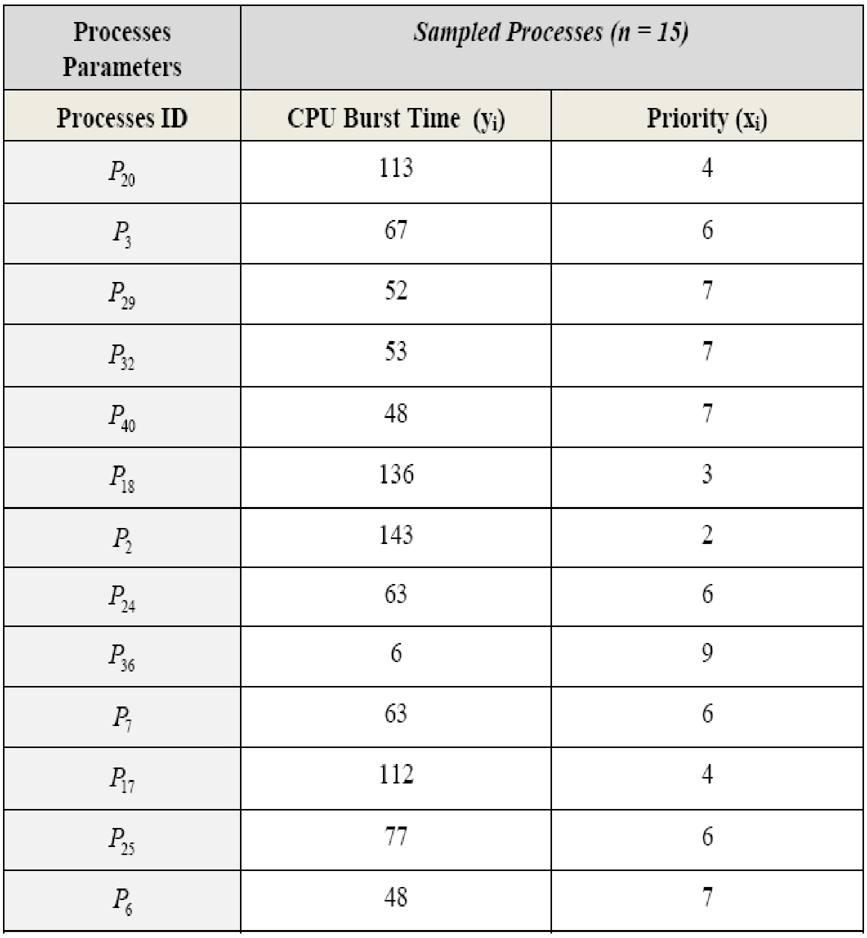
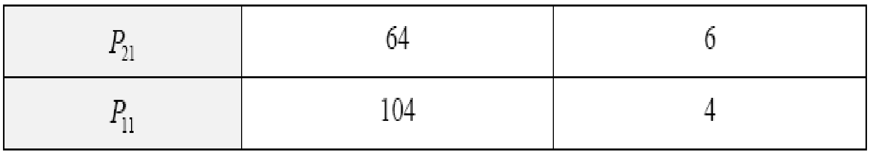
Processes Parameters	*Sampled Processes (n = 5)*				
Processes (PID)	P_1	P_8	P_{12}	P_{13}	P_{17}
CPU Burst Time (Y_i)	80	115	183	106	112
Priority (X_i)	5	4	0	4	4

Table 3: Description of second sample n = 10.



Processes Parameters	*Sampled Processes (n = 10)*	
Processes ID	CPU Burst Time (y_i)	Priority (x_i)
P_{20}	113	4
P_2	143	2
P_{28}	64	6
P_3	130	3
P_{39}	74	6
P_{22}	58	7
P_{10}	159	2
P_{15}	57	7
P_{17}	112	4
P_{23}	89	8

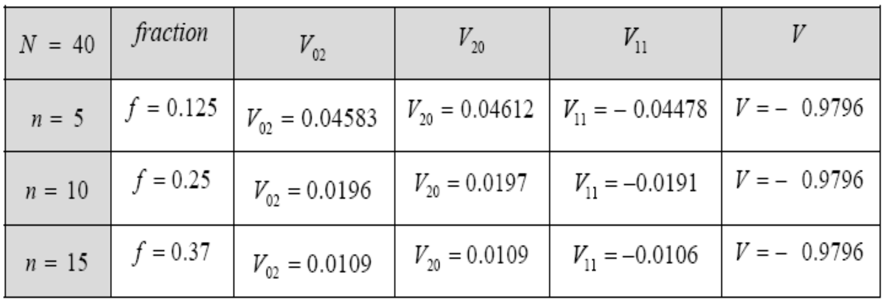
Table 4. Description of third sample n = 15.

Processes Parameters	*Sampled Processes (n = 15)*	
Processes ID	CPU Burst Time (y_i)	Priority (x_i)
P_{20}	113	4
P_3	67	6
P_{29}	52	7
P_{32}	53	7
P_{40}	48	7
P_{18}	136	3
P_2	143	2
P_{24}	63	6
P_{36}	6	9
P_7	63	6
P_{17}	112	4
P_{25}	77	6
P_6	48	7
P_{21}	64	6
P_{11}	104	4

The table 5, shows the ready queue process parameters for the first sample (n = 5), second sample (n = 10) and the third sample (n = 15) (from **Appendix A**).

Table 5. Ready Queue process parameters for (n = 5, 10 and 15).

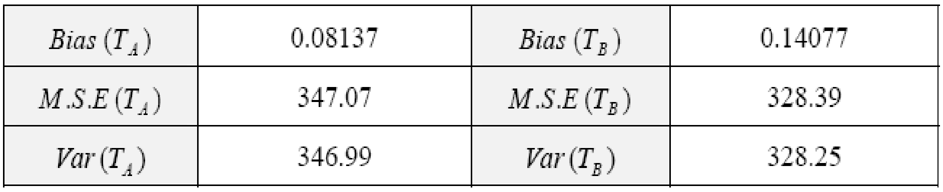


$N = 40$	*fraction*	V_{02}	V_{20}	V_{11}	V
$n = 5$	$f = 0.125$	$V_{02} = 0.04583$	$V_{20} = 0.04612$	$V_{11} = -0.04478$	$V = -0.9796$
$n = 10$	$f = 0.25$	$V_{02} = 0.0196$	$V_{20} = 0.0197$	$V_{11} = -0.0191$	$V = -0.9796$
$n = 15$	$f = 0.37$	$V_{02} = 0.0109$	$V_{20} = 0.0109$	$V_{11} = -0.0106$	$V = -0.9796$

5. PERFORMANCE MEASURES OF T_{TFT} (AS PER [18])

An estimator is said to be more efficient estimator of the parameter (i.e. ready queue length prediction) than the other provided that it produces lower M.S.E values and hence lower values of variances than the other one. Table 6, Table 7 and Table 8 presents numerical comparisons between the two estimators in terms of Bias and M.S.E.

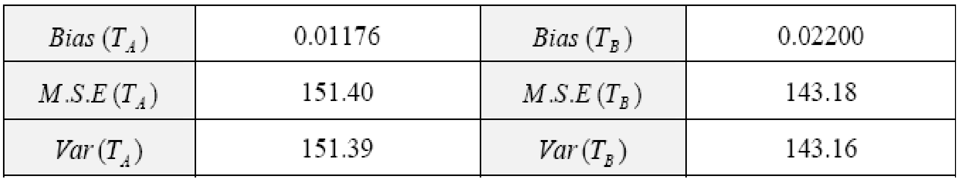
Table 6. Bias and M.S.E comparison of T_A and T_B estimators (when n = 5).



$Bias\,(T_A)$	0.08137	$Bias\,(T_B)$	0.14077
$M.S.E\,(T_A)$	347.07	$M.S.E\,(T_B)$	328.39
$Var\,(T_A)$	346.99	$Var\,(T_B)$	328.25

The M.S.E (= 328.39) and hence Variance (= 328.25) of T_B is lesser than the M.S.E (= 347.07) and Variance (= 328.39) of T_A in above example Table 6 when n = 5.

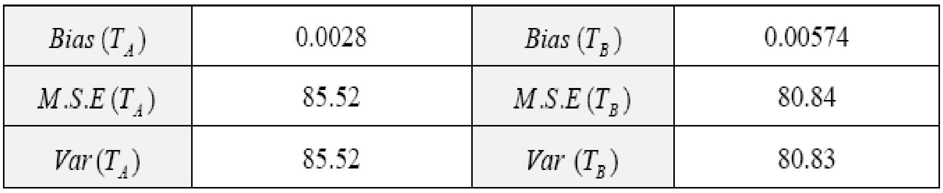
Table 7. Bias and M.S.E comparison of T_A and T_B estimators (when n = 10).



$Bias\,(T_A)$	0.01176	$Bias\,(T_B)$	0.02200
$M.S.E\,(T_A)$	151.40	$M.S.E\,(T_B)$	143.18
$Var\,(T_A)$	151.39	$Var\,(T_B)$	143.16

Similarly, the M.S.E (= 143.18) and hence Variance (= 143.16) of T_B is lesser than the M.S.E (= 151.40) and Variance (= 151.39) of T_A in above example Table 7 when n = 10.

Table 8. Bias and M.S.E comparison of T_A and T_B estimators (when n = 15).



$Bias\,(T_A)$	0.0028	$Bias\,(T_B)$	0.00574
$M.S.E\,(T_A)$	85.52	$M.S.E\,(T_B)$	80.84
$Var\,(T_A)$	85.52	$Var\,(T_B)$	80.83

Again it has been observed that the M.S.E (= 80.84) and Variance (= 80.83) of T_B is lesser than the M.S.E (= 85.52) and Variance (= 85.52) of T_A in above example Table 8 when n = 15. It has been observed that the estimator T_B produces better prediction for any values of *n*.

The 95% confidence interval of the mean time estimate using T_A and T_B are:

$$P[T_A \pm t_{n-1,\frac{\alpha}{2}} \sqrt{V(T_A)}] = 0.95 \quad ...(12)$$

$$P[T_B \pm t_{n-1,\frac{\alpha}{2}} \sqrt{V(T_B)}] = 0.95 \quad ...(13)$$

where

$$V(T_A) = \left[M.S.E(T_A) - \left(Bias(T_A)^2\right)\right] \quad ...(14)$$

$$V(T_B) = \left[M.S.E(T_\mathbf{B}) - \left(Bias(T_\mathbf{B})^2\right)\right] \quad ...(15)$$

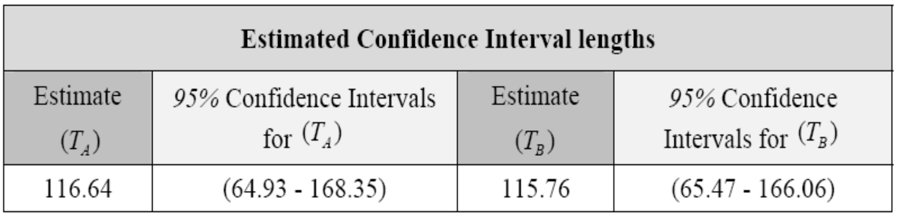
More explicitly, one can write 95% confidence intervals for mean estimation as

$$P[T_A - t_{n-1,\frac{\alpha}{2}} \sqrt{V(T_A)} \le \bar{Y} \le T_A + t_{n-1,\frac{\alpha}{2}} \sqrt{V(T_A)}] = 0.95$$

$$P[T_B \, - \, t_{n \, - \, 1, \frac{\alpha}{2}} \sqrt{V\left(T_B\right)} \, \leq \, \bar{Y} \, \leq \, T_B$$
$$+ \, t_{n \, - \, 1, \frac{\alpha}{2}} \sqrt{V\left(T_B\right)}] \, = \, 0.95$$

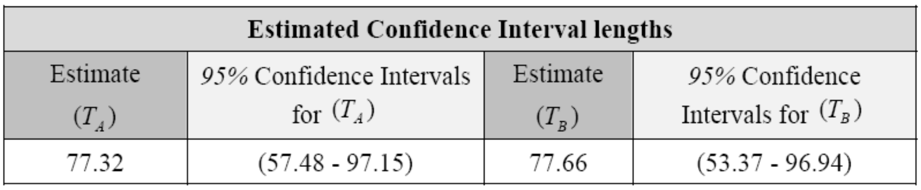
where $t_{n \, - \, 1, \frac{\alpha}{2}}$ is t-value at (n - 1) degree of freedom and at α -level of significance. The T_A , T_B are the sample based estimates of ready queue processing time parameter $\bar{Y}$. The t values are shown in the **Appendix B**. The time estimates T_A , T_B are predictors in the form of average time required to complete a processes by processors using lottery scheduling. Table 9, Table 10 and Table 11 shows comparisons of the two time estimates and confidence intervals for the ready queue length in case of first sample (n = 5), second sample (n = 10) and third sample (n = 15) respectively.

Table 9. Confidence interval comparison of T_A and T_B estimators (when n = 5).



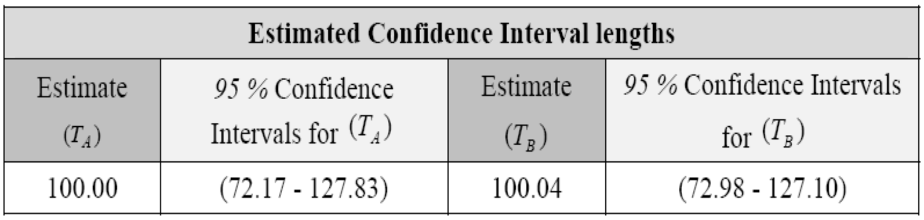
Estimated Confidence Interval lengths			
Estimate (T_A)	95% Confidence Intervals for (T_A)	Estimate (T_B)	95% Confidence Intervals for (T_B)
116.64	(64.93 - 168.35)	115.76	(65.47 - 166.06)

Table 10. Confidence interval comparison of T_A and T_B estimators (when n = 10).



Estimated Confidence Interval lengths			
Estimate (T_A)	95% Confidence Intervals for (T_A)	Estimate (T_B)	95% Confidence Intervals for (T_B)
77.32	(57.48 - 97.15)	77.66	(53.37 - 96.94)

Table 11. Confidence interval comparison of T_A and T_B estimators (when n = 15).



Estimated Confidence Interval lengths			
Estimate (T_A)	95 % Confidence Intervals for (T_A)	Estimate (T_B)	95 % Confidence Intervals for (T_B)
100.00	(72.17 - 127.83)	100.04	(72.98 - 127.10)

These time estimates, hence confidence intervals as per Table 9 to11 shows that estimator (T_B) consistently produced narrow time intervals than estimator (T_A) .

So, the predicted total time required for processing remaining jobs in the ready queue using two methods are:

Remaining time estimate by

$$t_A : t_A \, = \, \left(N \, - n\right) \, T_A \qquad ....(16)$$

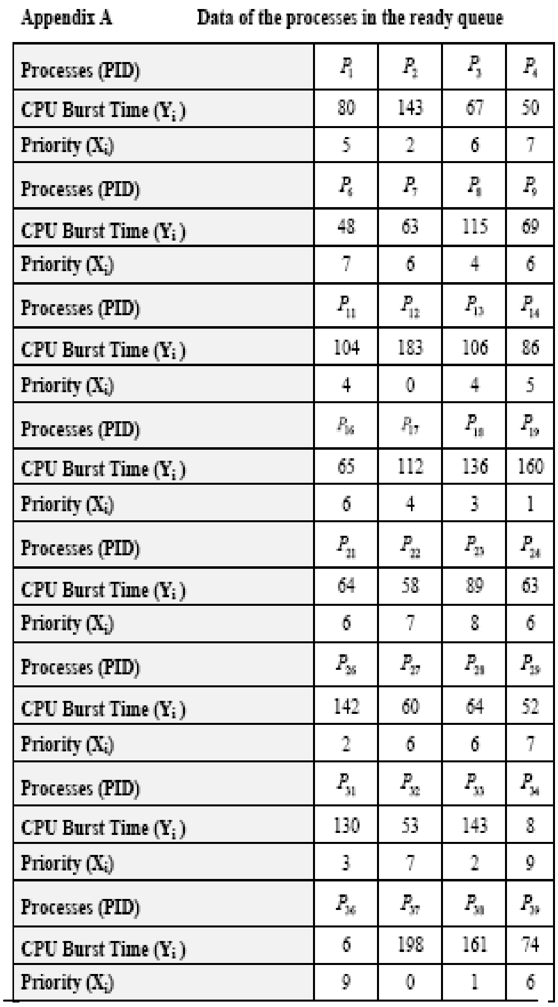
Remaining time estimate by t_B **:**

$$t_B \, = \, \left(N \, - \, n\right) \, T_B \qquad ...(17)$$

6. DISCUSSION AND CONCLUSION

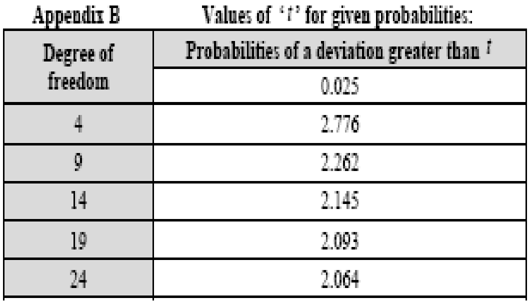
This paper considers, two ready queue processing time estimation methods T_A and T_B which are mutually compared. In given examples the estimator T_B is uniformly efficient over estimator T_A for n = 5, 10 and 15, as shown in Table 9, Table 10 and Table 11 respectively (due to lower M.S.E). In each case, true values of CPU burst time lies within the range of confidence intervals when we use *(T-F-T)* estimator.

Incorporation of lottery scheduling in ready queue processing time estimate guarantees equal chance of selection of processes to the processor, to avoid starvation. The proposed estimators may be incorporated in practical utility programs for performance analysis or billing purpose. So, it is recommended to prefer T_B over T_A to predict better about remaining processing time. In the setup of lottery scheduling when variables are negatively correlated, for the ready queue Length predictions, such estimators are extremely useful.



Appendix A Data of the processes in the ready queue

Processes (PID)	P_1	P_2	P_3	P_4
CPU Burst Time (Y_i)	80	143	67	50
Priority (X_i)	5	2	6	7
Processes (PID)	P_6	P_7	P_8	P_9
CPU Burst Time (Y_i)	48	63	115	69
Priority (X_i)	7	6	4	6
Processes (PID)	P_{11}	P_{12}	P_{13}	P_{14}
CPU Burst Time (Y_i)	104	183	106	86
Priority (X_i)	4	0	4	5
Processes (PID)	P_{16}	P_{17}	P_{18}	P_{19}
CPU Burst Time (Y_i)	65	112	136	160
Priority (X_i)	6	4	3	1
Processes (PID)	P_{21}	P_{22}	P_{23}	P_{24}
CPU Burst Time (Y_i)	64	58	89	63
Priority (X_i)	6	7	8	6
Processes (PID)	P_{26}	P_{27}	P_{28}	P_{29}
CPU Burst Time (Y_i)	142	60	64	52
Priority (X_i)	2	6	6	7
Processes (PID)	P_{31}	P_{32}	P_{33}	P_{34}
CPU Burst Time (Y_i)	130	53	143	8
Priority (X_i)	3	7	2	9
Processes (PID)	P_{36}	P_{37}	P_{38}	P_{39}
CPU Burst Time (Y_i)	6	198	161	74
Priority (X_i)	9	0	1	6



Appendix B Values of 't' for given probabilities:

Degree of freedom	Probabilities of a deviation greater than t
	0.025
4	2.776
9	2.262
14	2.145
19	2.093
24	2.064

7. REFERENCES

[1] A.C. Waldspurger, E.W. Weihl, "Lottery Scheduling a flexible proportional-share resource management", Proceedings of the 1st USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp.1-11,1994.

[2] D. Petro, A.G. Gibson, W.J. Milford, "Implementing Lottery Scheduling: Matching the specializations in Traditional Schedulers", Proceedings of the USENIX Annual Technical Conference USA, pp.66-80, 1999.

[3] D. Yiping, F. William, "Interpreting Windows NT Processor Queue Length Measurements", Proceedings of the 31st Computer Measurement Group Conference, Vol. 2, pp.759-770, 2000.

[4] A. Tanenbaum, Operating System, Ed. 8, Prentice Hall of India, New Delhi, 2000.

[5] A. Silberschatz, P. Galvin, Operating System Concepts, Ed.5, John Wiley and Sons (Asia), Inc, 1999.

[6] Cochran, Sampling Technique, Wiley Eastern Publication, New Delhi, 2005.

[7] D. Singh, F.S. Choudhary, Theory and Analysis of Sample Survey and Designs, Wiley Eastern Limited, New Delhi, 1986.

[8] D. Shukla, A. Jain, A. Chowdhary, " Estimation of ready queue processing time under Usual Lottery Scheduling (ULS) scheme in multiprocessor environment", Journal of Applied Computer Science and Mathematics (JACSM), Vol. 11, no. 11, pp.58-63, 2011.

[9] D. Shukla, A. Jain, "Estimation of ready queue processing time under SL-Scheduling scheme in multiprocessor environment", International Journal of Computer Science and Security (IJCSS), Vol. 4(1), pp. 74-81, 2010.

[10] D. Shukla, A. Jain, A. Chowdhary," Estimation of ready queue processing time under Usual Group Lottery Scheduling (GLS) scheme in multiprocessor environment", International Journal of Computer and Applications (IJCA), Vol. 8, no. 14, pp. 39-45, 2010.

[11] D. Raz, B. Itzahak, H. Levy, "Classes, Priorities and Fairness in Queuing Systems", Research report, Rutgers University, 2004.

[12] D. Shukla, A. Jain, "Analysis of ready queue processing time under PPS-LS and SRS-LS scheme in multiprocessing environment", GESJ: Computer Science and Telecommunication, Vol.33, no.1, pp. 54-61, 2012.

[13] D. Shukla, A. Jain, "Estimation of ready queue processing time using Efficient Factor -Type (E-F-T) estimator in multiprocessor environment", International Journal of Computer and Applications (IJCA), Vol. 48, no. 16, pp. 20-27, 2012.

[14] V.K. Singh, D. Shukla, "One parameter family of factor type ratio estimators", METRON International Journal of Statistics, Vol. XLV- no. 1-2, pp. 273-283, 1987.

[15] V.K. Singh, D. Shukla,"An efficient one-parameter family of factor-type estimator in sample surveys", METRON International Journal of Statistics, Vol. XVV, pp.139-159, 1992.

[16] W. Stalling, Operating Systems, Ed.5, Pearson Education, Singapore, Indian Edition, New Delhi, 2004.

[17] T. Srivenkataramana, "A dual to ratio estimator in sample survey", Biometrika, Vol. 67, pp. 199-204, 1980.

[18] D. Shukla, V.K. Singh, G.N. Singh, "On the use of transformation in factor-type estimator", METRON International Journal of Statistics, Vol. XLV, pp. 349-361, 1993.