<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>80</td>
</tr>
<tr>
<td>4.2</td>
<td>Numerical Example 1</td>
<td>80</td>
</tr>
<tr>
<td>4.3</td>
<td>Numerical Example 2</td>
<td>83</td>
</tr>
<tr>
<td>4.4</td>
<td>Fork Join Queue System</td>
<td>85</td>
</tr>
<tr>
<td>4.5</td>
<td>Splitting Queue and a Flexible Distributed Simulation System</td>
<td>92</td>
</tr>
</tbody>
</table>
CHAPTER 4: PRACTICAL APPLICATIONS

4.1 INTRODUCTION

Here all the theories are tested through some mathematical examples for practical situation. Here parameters in the mathematical model selected in such a way that they preserve situation for the continuation of a stationary process as well as the stability conditions. One mathematical software package of Matlab is used in order to create the results.

4.2 NUMERICAL EXAMPLE 1:

Consider the Model 1, as explained in section 3.3.1, for both buffers finite. In this numerical, $M_1 = 7$ & $M_2 = 3, 8 & 13$, with subsequent values of constraint along with probabilities were used.

Table 4.2.1: Numerical Results for Model 1

<table>
<thead>
<tr>
<th>$\lambda_1$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\rho_1$</th>
<th>$\rho_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>262.50</td>
<td>237.50</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>15</td>
<td>393.75</td>
<td>356.25</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>25</td>
<td>437.50</td>
<td>395.83</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>35</td>
<td>574.22</td>
<td>461.81</td>
<td>0.80</td>
<td>0.90</td>
</tr>
<tr>
<td>45</td>
<td>590.63</td>
<td>356.25</td>
<td>1.00</td>
<td>1.50</td>
</tr>
<tr>
<td>55</td>
<td>577.50</td>
<td>326.56</td>
<td>1.25</td>
<td>2.00</td>
</tr>
<tr>
<td>65</td>
<td>426.56</td>
<td>308.75</td>
<td>2.00</td>
<td>2.50</td>
</tr>
<tr>
<td>75</td>
<td>393.75</td>
<td>296.88</td>
<td>2.50</td>
<td>3.00</td>
</tr>
<tr>
<td>85</td>
<td>371.88</td>
<td>288.39</td>
<td>3.00</td>
<td>3.50</td>
</tr>
<tr>
<td>95</td>
<td>356.25</td>
<td>282.03</td>
<td>3.50</td>
<td>4.00</td>
</tr>
</tbody>
</table>

Also the values of the additional parameters were:
\[ p_1^1 = 0.3, \quad q_1 = 0.20, \quad p_1^2 = 0.3, \quad p_{S_1}^1 = 0.20, \quad q_{S_1} = 0.20, \quad p_{S_2}^1 = 0.40, \quad p_{S_2}^2 = 0.40, \quad q_2 = 0.20, \quad p_2^1 = 0.30, \quad p_2^{S_2} = 0.20, \quad q_{S_2} = 0.20, \quad p_{S_2}^1 = 0.40, \quad p_{S_2}^2 = 0.40. \]

If the values of \( M_1 \) and \( M_2 \) are: \( M_1 = 7, \quad M_2 = 13 \) among the distinct rates of \( \lambda, \mu_1 \) and \( \mu_2 \); the following can be obtained.

If \( \lambda = 5 \) then \( \mu_1 = 262.50, \quad \mu_2 = 237.50 \)

Probability for structure being idle : \( P_{0,0} = 0.6989 \)

Probability for the first station being idle : \( \sum_{j=0}^{M_2} P_{0,j} = 0.7786 \)

Probability that the second station being idle : \( \sum_{i=0}^{M_1} P_{i,0} = 0.7037 \)

Probability that the first station being blocked : \( \sum_{i=1}^{M_1} P_{i,M_2+1} = 0.0012 \)

Probability that the second station being blocked : \( \sum_{j=1}^{M_2} P_{M_1+1,j} = 0.0064 \)

Probability that the first station being busy : \( \sum_{j=1}^{M_2} P_{0,j} = 0.2214 \)

Probability that the second station being busy : \( \sum_{i=0}^{M_1} P_{i,0} = 0.2205 \)

Probability for an outer incoming task is missing, i.e., while first position is packed or second position is blocked : \( \sum_{j=0}^{M_2+1} P_{M_1,j} + \sum_{j=1}^{M_2} P_{M_1+1,j} = 0.0134. \)

Also when If \( \lambda = 95 \) then \( \mu_1 = 356.25, \quad \mu_2 = 282.03 \)

Possibility for structure becomes inactive: \( P_{0,0} = 0.0010 \)

Probability that the first station being idle : \( \sum_{j=0}^{M_2} P_{0,j} = 0.0238 \)

Probability that the second station being idle : \( \sum_{i=0}^{M_1} P_{i,0} = 0.0554 \)
Probability that the first station being blocked: $\sum_{i=1}^{M_1} P_{i,M_2+1} = 0.1362$

Probability that the second station being blocked: $\sum_{j=1}^{M_2} P_{M_1+1,j} = 0.2065$

Probability that the first station being busy: $\sum_{j=1}^{M_2} P_{0,j} = 0.9762$

Probability that the second station being busy: $\sum_{i=0}^{M_1} P_{i,0} = 0.9703$

Probability of outer incoming mission is missing: $\sum_{j=0}^{M_2+1} P_{M_1,j} + \sum_{j=1}^{M_2} P_{M_1+1,j} = 0.4580$.

If rate of $\lambda$ increase, percentage of arrivals quantity for structure becomes similar. Here both the locations are hectic having equal percentage of times either among rate of $\lambda$ or several discrepancies happen. The variation for idle times is not same also combined possibility allocation for given mission during first station & second station at $\lambda= 5m$; $M_1 = 30$ and $M_2 = 3$ may be known as follows:

- $P_{0,0} = 0.0054$, $P_{0,1} = 0.0036$, $P_{0,2} = 0.0043$, $P_{0,3} = 0.0052$
- $P_{1,0} = 0.0044$, $P_{1,1} = 0.0043$, $P_{1,2} = 0.0052$, $P_{1,3} = 0.0063$
- $P_{1,4} = 0.0044$, $P_{2,0} = 0.0054$, $P_{2,1} = 0.0059$, $P_{2,2} = 0.0076$
- $P_{2,3} = 0.0097$, $P_{2,4} = 0.0096$, $P_{3,0} = 0.0075$, $P_{3,1} = 0.0088$
- $P_{3,2} = 0.0115$, $P_{3,3} = 0.0148$, $P_{3,4} = 0.0165$, $P_{4,0} = 0.0113$
- $P_{4,1} = 0.0137$, $P_{4,2} = 0.0175$, $P_{4,3} = 0.0219$, $P_{4,4} = 0.0258$
- $P_{5,0} = 0.0182$, $P_{5,1} = 0.0218$, $P_{5,2} = 0.0259$, $P_{5,3} = 0.0310$
- $P_{5,4} = 0.0381$, $P_{6,0} = 0.0313$, $P_{6,1} = 0.0341$, $P_{6,2} = 0.0345$
- $P_{6,3} = 0.0438$, $P_{6,4} = 0.0548$, $P_{7,0} = 0.0606$, $P_{7,1} = 0.0410$
- $P_{7,2} = 0.0393$, $P_{7,3} = 0.0263$, $P_{7,4} = 0.1422$, $P_{8,1} = 0.0487$
- $P_{8,2} = 0.0467$, $P_{8,3} = 0.0313$

### 4.3 Numerical Example 2:
Consider the second Model in which first buffer infinite and second buffer is finite. Let $M_1 = 300, M_2 = 3, 8 & 13$. Highest values of $M_1$ guarantee that the endless ability for every possibility is zero. In current example during order to satisfy stability condition given by equation (3.3.17) let

$$\sigma_1 = q_1 + p_1^2 - p_1^S p_{S_1}^1 \quad \text{and} \quad \sigma_2 = p_2^1 + p_2^S p_{S_2}^1.$$ 

Then all the probabilities and parameters were chosen such that

$$q_1 + p_1^S + p_1^2 + p_1^1 = 1, \quad q_{S_1} + p_{S_1}^1 + p_{S_1}^2 = 1, \quad q_2 + p_2^1 + p_2^S = 1, \quad q_{S_2} + p_{S_2}^1 + p_{S_2}^2 = 1,$$

$$\sigma_1 > 0 \quad \text{and} \quad \lambda < \sigma_2 \mu_2 \pi_0 + (\sigma_1 \mu_1 - \sigma_2 \mu_2) (1 - \pi_{M_2+1}) .$$

The following values of the parameters and probabilities were selected.

**Table 4.3.1: Numerical Results for Model 2**

<table>
<thead>
<tr>
<th>$\lambda_1$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\rho_1$</th>
<th>$\rho_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>535.71</td>
<td>188.78</td>
<td>0.05</td>
<td>0.25</td>
</tr>
<tr>
<td>10</td>
<td>535.71</td>
<td>235.97</td>
<td>0.10</td>
<td>0.40</td>
</tr>
<tr>
<td>15</td>
<td>535.71</td>
<td>283.16</td>
<td>0.15</td>
<td>0.50</td>
</tr>
<tr>
<td>20</td>
<td>535.71</td>
<td>314.63</td>
<td>0.20</td>
<td>0.60</td>
</tr>
<tr>
<td>25</td>
<td>446.43</td>
<td>337.10</td>
<td>0.30</td>
<td>0.70</td>
</tr>
<tr>
<td>30</td>
<td>321.43</td>
<td>377.55</td>
<td>0.50</td>
<td>0.75</td>
</tr>
<tr>
<td>35</td>
<td>340.391</td>
<td>412.95</td>
<td>0.55</td>
<td>0.80</td>
</tr>
<tr>
<td>40</td>
<td>357.14</td>
<td>444.18</td>
<td>0.60</td>
<td>0.85</td>
</tr>
<tr>
<td>45</td>
<td>344.39</td>
<td>471.94</td>
<td>0.70</td>
<td>0.90</td>
</tr>
<tr>
<td>50</td>
<td>357.14</td>
<td>471.91</td>
<td>0.75</td>
<td>1.00</td>
</tr>
</tbody>
</table>

$p_1^1 = 0.1, \quad q_1 = 0.10, \quad p_1^2 = 0.60, \quad p_1^S = 0.20, \quad q_{S_1} = 0.20, \quad p_{S_1}^1 = 0.10, \quad p_{S_1}^2 = 0.30, \quad p_2^1 = 0.30, \quad q_2 = 0.30, \quad p_2^2 = 0.20, \quad p_2^S = 0.20, \quad q_{S_2} = 0.20, \quad p_{S_2}^1 = 0.40, \quad p_{S_2}^2 = 0.40.$
Now consider the case with $M_1 = 300, M_2 = 13$ among two rates of $\lambda, \mu_1$ and $\mu_2$, then the following are obtained.

If $\lambda = 5$ then $\mu_1 = 262.50, \mu_2 = 237.50$

Probability for structure being idle : $P_{0,0} = 0.7405$

Probability that the first station being idle : $\sum_{j=0}^{M_2} P_{0,j} = 0.9494$

Probability that the second station being idle : $\sum_{i=0}^{M_1} P_{i,0} = 0.7405$

Probability that the first station being blocked : $\sum_{i=1}^{M_1} P_{i,M_2+1} = 0.0015$

Probability that the first station being busy : $\sum_{j=1}^{M_2} P_{0,j} = 0.0506$

Probability that the second station being busy : $\sum_{i=0}^{M_1} P_{i,0} = 0.0360$.

If $\lambda = 50$ then $\mu_1 = 356.25, \mu_2 = 282.03$

Probability for structure becomes idle : $P_{0,0} = 0.1003$

Probability that the first station being idle : $\sum_{j=0}^{M_2} P_{0,j} = 0.3152$

Probability that the second station being idle : $\sum_{i=0}^{M_1} P_{i,0} = 0.1004$

Probability that the first station being blocked : $\sum_{i=1}^{M_1} P_{i,M_2+1} = 0.0604$

Probability that the first station being busy : $\sum_{j=1}^{M_2} P_{0,j} = 0.6848$

Probability that the second station being busy : $\sum_{i=0}^{M_1} P_{i,0} = 0.6146$

4.4. FORK JOIN QUEUE SYSTEM
Most important application of the fork join queue system is in advance processor as well as message structures which require everyday high-superiority representations in order to the direction of helping forecasting presentation behavior for a broad collection of area. In real life responsibility of structures know how to distract when an employment flow, for that every employment is divide in a lot of synchronized missions which are developed similar at a range of probably mixed, models. This kind of Instances schemes including compact disk, industrialized systems and multi programming.

![Figure 4.4.1 Fork Joint Queuing Model](image)

Theoretically, this kind of structures is invented as fork-join line inside a blocked queuing system as shown within above Fig. 4.1. In a fork-join queuing structure, every incoming job is dividing into to \( N \) tasks, which queue in order to repair previous to linking a line.

At this time too much difficult for representation moment of job response time in fork-join synchronization analytically. The repair instance for disc constrain be dependent lying on diskette tube look for instance along with revolving regulations which is improbable scattered finitely. Therefore, disc displays needs fork-join representation through M/G/1 equivalent line up. Lastly, a disc collection could assist up to 50 disc drives thus the logical estimate requires to be competent of create consequences rapidly for a many disks. In turn to resolve fork-join queuing networks logically, the majority consequences suppose that the similar queues are self-
determining and uniformly spread. The entrance time for split is $\lambda$ as well as repair speed at every line is $\mu$. By administration imitations for the fork-join system through dissimilar assessments for $N$, indicate reply instance estimate is standardizing for subsequent outcome:

$$R_N \approx \left[ \frac{H_N}{H_2} + \frac{4}{11} \left( 1 - \frac{H_N}{H_2} \right) \rho \right] \left( \frac{12 - \rho}{8} \right) \frac{1}{\mu(1 - \rho)} \quad N \geq 2$$

(4.4.1)

Where $H_N$ is the harmonic series: $\sum_{i=1}^{N} \frac{1}{i}$.

Here exclamation involving normal as well as full travel on behalf of estimating indicates reaction instance of MI/G/1 lines:

$$R_N \approx \left[ H_N + \left( \sum_{i=1}^{N} \binom{N}{i} (-1)^{i-1} \sum_{m=1}^{i} \binom{i}{m} \frac{(m-1)!}{i^{m+1}} \right) - H_N \right] \frac{\lambda}{\mu} \frac{1}{\mu - \lambda} \quad 0 \leq \lambda < \mu \quad N \geq 2$$

(4.4.2)

Above outcome may also generalized in the direction of arrival times and non-exponential service, but it is applicable for uniform servers.

Another approximation can be given for the same conditions as equation (4.4.1) and (4.4.2)

$$R_N \approx \frac{1}{\mu} \left( H_N + \frac{\rho}{2(1 - \rho)} \left( \sum_{i=1}^{N} \frac{1}{i - \rho} + (1 - 2\rho) \sum_{i=1}^{N} \frac{1}{i(i - \rho)} \right) \right)$$

(4.4.3)

David (1981) explains an upper bound for the mean of the highest of a set of n random variables, $X_i$.

$$E[X_{(n)}] \leq \mu + \frac{\sigma(n - 1)}{\sqrt{2n - 1}}$$

(4.4.4)
Here $\sigma$ be standard deviation of $X$ with $\mu$ as average. In 1994 Tantaw & Thomasian accept equation (4.4.4). From Melton and Tanami’s technique for examine reproduction; where estimation recommended:

$$R_N(\rho) \approx R_1(\rho) + \sigma_1(\rho)F_N\alpha_N(\rho)$$

(4.4.5)

$\sigma_1(\rho)$ and $R_1(\rho)$ are respectively the standard deviation and mean response time designed for fork join belongings. $F_N$ be constant and $\alpha_N(\rho)$ is a scales according to observations from simulation results. Both give approximation to fork-join synchronization, by modeling a parallel network known as the split-merge model. (Figure 4.2). Basically as soon as each jobs complete their services and rejoin can the next job split into tasks and start servicing. Above discussion guide in the direction of normal average reply times as compared with given fork-join correspondent.

Let $f_n(\alpha, t)$ represents a probability function which denotes the highest of $n$ independent, depressing exponential arbitrary variables, having parameters $\alpha = (\alpha_1, \ldots, \alpha_n)$. The below mentioned formula is derived in favor of Laplace transform for $f_n(\alpha, t), L_n(\alpha, s)$.

$$\left(s + \sum_{j=1}^{m} \alpha_j\right) L_m(\alpha, s) = \sum_{j=1}^{m} \alpha_j L_{m-1}(\alpha \setminus j, s) \quad ; s \geq 0$$

(4.4.6)
For 1 \leq m \leq n, where \(\alpha = (\alpha_1, \ldots, \alpha_{j-1}, \alpha_{j+1}, \ldots, \alpha_m)\), \(L_0(\epsilon, s) = 1\) where \(\epsilon\) is the zero vector.

Here \(k^{th}\) moments, \(M_n(\alpha, k)\) for \(f_n(\alpha, t)\) can be derived by differentiating (4.4.6) \(k\) times by Leibnitz’s result and replacing \(s\) to zero. Here the relation between them is given as under:

\[
I(k, \alpha, M) = \frac{1}{k} \sum_{i=1}^{k} I(k - 1, \alpha_{\backslash i}, M_{\backslash i}) + \alpha_i M_i L_{k-1}(\alpha_{\backslash i}, \alpha_i)/2 \quad I(1, \alpha, M_1) = 1/\alpha_1
\]

(4.4.7)

The formula for Highest categorize value is given by:

\[
F_{X_{(n)}}(x) = P(X_{(n)} \leq x) ; \forall i = P(X_{(i)} \leq x)
\]

If \(X_1, X_2, \ldots, X_n\) are iid with cumulative distribution function (cdf) \(F(x)\);

\[
F_{X_{(n)}}(x) = (F(x))^n
\]

Now cdf of \(X_i\) is \(F_i(x)\) then;

\[
F_{X_{(n)}}(x) = \prod_{i=1}^{n} F_i(x)
\]

Here average for highest \(n\) free arbitrary inconsistent followed by

\[
E[X_{(n)}] = \int_{-\infty}^{\infty} x \left( \sum_{i=1}^{n} \frac{f_i(x)}{F_i(x)} \right) \prod_{i=1}^{n} F_i(x) dx
\]

(4.4.8)

If the random variables are iid, (4.4.8) can be simplified as
Further moments $M_k$, can be obtained as

$$M_k = E\left[X_{(n)}^k\right] = n \int_{-\infty}^{\infty} x^k f(x)(F(x))^{(n-1)} dx \quad (4.4.10)$$

The solution of the mean of the maximum random variables can be exactly obtained these formulations, irrelevant for the allocation of the arbitrary variables.

Here Table 4.4.1 compares techniques as well as reproduction calculations of Erlang-$k$ allocation, having constraint $k$. In the table, column of $HZ$ includes a result and column of $OS$ includes the results obtained from equation (4.4.9). Here estimate go through small discrepancy since $N \to \infty$, for an incessantly rising proportion inaccuracy of the higher value of $N$. Equation (4.4.9) constantly send improved percentage errors with no clear presentation discrepancy.

**Table 4.4.1 : Comparison with Erlang with low Variance**

<table>
<thead>
<tr>
<th></th>
<th>Erlang - 4</th>
<th>HZ</th>
<th>Sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>% err</td>
<td>% err</td>
<td>% err</td>
<td>% err</td>
</tr>
<tr>
<td>0.060</td>
<td>1.000</td>
<td>0.999</td>
<td>0.999</td>
</tr>
<tr>
<td>6.127</td>
<td>1.273</td>
<td>1.195</td>
<td>1.195</td>
</tr>
<tr>
<td>13.993</td>
<td>1.969</td>
<td>1.380</td>
<td>1.380</td>
</tr>
<tr>
<td>16.820</td>
<td>2.339</td>
<td>1.555</td>
<td>1.555</td>
</tr>
<tr>
<td>26.430</td>
<td>3.6300</td>
<td>1.716</td>
<td>1.716</td>
</tr>
</tbody>
</table>
If there is a very large difference condition, a Pareto distribution can be used. Also table 4.4.2 compares the models as well as reproduction consequences on behalf of weighty tailed Pareto $\beta$ distribution. It have cdf:

$$F_p(x) = 1 - \alpha(x + \gamma)^{-\beta} \quad \text{with} \quad \alpha = \gamma^\beta \quad \text{and} \quad \gamma = \beta - 1.$$
These results are too much improved for a highest categorize value of estimation along with supplementary precise in the case of the small inconsistency.

**Table 4.4.2 Comparison with Pareto with high variance**

<table>
<thead>
<tr>
<th></th>
<th>OS</th>
<th>HZ</th>
<th>Sim</th>
<th>OS</th>
<th>HZ</th>
<th>Sim</th>
<th>OS</th>
<th>HZ</th>
<th>Sim</th>
<th>OS</th>
<th>HZ</th>
<th>Sim</th>
<th>OS</th>
<th>HZ</th>
<th>Sim</th>
<th>OS</th>
<th>HZ</th>
<th>Sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td>0.614</td>
<td>-0.707</td>
<td>-0.132</td>
<td>-0.001</td>
<td>-0.096</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>1.000</td>
<td>1.556</td>
<td>2.266</td>
<td>3.129</td>
<td>4.149</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td>0.614</td>
<td>6.350</td>
<td>7.774</td>
<td>5.173</td>
<td>0.512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sim</td>
<td>1.000</td>
<td>1.667</td>
<td>2.444</td>
<td>3.290</td>
<td>4.174</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td></td>
<td>1.556</td>
<td>2.266</td>
<td>2.266</td>
<td>1.556</td>
<td>-0.001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td></td>
<td>1.000</td>
<td>1.350</td>
<td>1.567</td>
<td>0.099</td>
<td>0.381</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td></td>
<td>1.000</td>
<td>1.350</td>
<td>1.567</td>
<td>0.099</td>
<td>0.381</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sim</td>
<td></td>
<td>1.000</td>
<td>1.567</td>
<td>2.694</td>
<td>1.667</td>
<td>0.509</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td></td>
<td>1.000</td>
<td>1.567</td>
<td>2.694</td>
<td>1.667</td>
<td>0.509</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td></td>
<td></td>
<td>1.000</td>
<td>1.567</td>
<td>2.694</td>
<td>1.667</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td></td>
<td></td>
<td>1.000</td>
<td>1.567</td>
<td>2.694</td>
<td>1.667</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sim</td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
<td>1.567</td>
<td>2.694</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
<td>1.567</td>
<td>2.694</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
<td>1.567</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
<td>1.567</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% err</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.5 SPLITTING QUEUE AND A FLEXIBLE DISTRIBUTED SIMULATION SYSTEM

We know that Simulation is the very oldest computer relevance: using a model exists for quasi-experimental learning is a cost-effective and is also intermittently only mode for study an actual structure and taking a base for conclusion. Its additional current reason is to make effective situation for guidance for an activity. There is a lot of range for modeling advancing and also for a calculation technique. Isolated occurrence simulation, which at presently our theme at this time focuses on the dynamics of the objective for the scheme as a chain of immediate situation transformations which is known as events. (A queuing technique is a traditional instance, through dealings like an employment incoming or departure from a queue.) In this concept, nothing occurs in between the incidence times, hence the simulation pass over the period. Calculation instance and simulated instance becomes independent thought: the calculation devote its time then also efforts for events—in authenticity, which have no time at all.

Above orthogonally contains mainly two important consequences. Out of these two; first is a computer simulation must be classify of a level which quicker or slower comparating the actual object; which occur on the regularity of procedures and their computational charge. Then after onwards second, to create simulate distance continue, in attendance should exist constantly upcoming incident in the direction of carry on. Incident dealing out has therefore supplementary force in the direction of decrease succeeding incidents through existing conditions. (For a queuing instance, the event of a job in coming service may involve leaving it in some specific time afterward.) Then after such types of incidents be subsequently control via respective incident record (schedule, timetable, planning, etc.), main concern line while match up planned in to a time-stamps, in which the leading entry decides the present incident as well as present instance.
Most important object for comparable in addition to spread reproduction; additional wide spread presentations also be found in J. Misra (1986), Fersha (1996) or Fujimota(2000) which is speed-up or, from a different angle it is seen that, the capacity to run superior models: after dividing a large portion of work into so many small parts which having the ground along with, we get outcomes earlier. This indicates that there must be sufficient incapacitate effort to compensate the extra attempts for skills and hence it is depends on the exact pattern under learning and also for the working manners. A spread simulation contains a set of Linear Programming, which all are represents a singular fraction of the queue system. There does not exist any worldwide status and the consistent techniques exclusively cooperate through a passing message.

These messages communicates of the occurrence notices in the list of events exists in the chronological model; over headset face, which locates via line in addition to effort for increasing time stamp sort, for which the beginning of the queue significant the limited virtual time in order to the receiving a Linear Programming. As there are so many queues as well as times for consistent techniques, the similarity through chronological side perform expand in entire image. Actually, within significance of rapidity, we may incapable to maintain every linear programming as similar occasion in instance as an only time hardly ever permits too much concurrency instead of any exploits. That reliance of occurrences of the any additional incidents, during representation generally a large amount meager related to a chronological sort. Hence common thought be simply nearby guarantees which all occurrences be practice within instance-squash arrange. However especially in favor of slackly joined techniques, this apparently normal obligation is difficulty; in the direction of practice the starting occurrence commencing its incoming line up; linear Programming should be familiar with the facts that it will never accept any occurrence message with a silent lesser time stamp.

Generally in sight main major techniques for trading by means of such problem are, one is “conservative” as well as following other is “optimistic” first (which is famous as “time warp”): within predictable advance, linear programming stay till there is a conventional communication via each linear programming which is corresponds through it. It is presumptuous so as to all communications comes via similar source enter in time stamp order, the linear programming then process safely the communication with negligible times. Thus waiting instruction, when it applies to each Linear Programming, is an exposed request to blocking. Actually there are so
many ways out. Unique opportunity is the calculation of so titled useless messages which is not appraising about measures but, in contrast, assurance their absenteeism up to a specified time; additional approach is to successively calculate harmless interval windows.

On the other hand, main approach of linear programming certainly not delays for probable additional communications continue over beginning of everything may be conventional so far. Currently “hopeful” performance is independent through deadlock, except goes the possibility for receiving onward a significant contribution and hence it creating errors. So when there is a consistent technique accepts input which has been reflected previous, it must be a rollback to an earlier condition in addition to terminate its unacceptable provisional production, as a result probably making rollbacks to another place. Obviously, both above clarifications have their own disadvantages, and they can be presented to be unfeasibly useless on malicious examples. In both the cases, numerous alternatives and adjustments has been suggested and calculated. Thus both the systems can be exposed to be truthful (in the sense that it make progress) just in a slightly nonstop way, however there is a precisely more complex and also informative substitute. The reliable techniques that create a spread simulation produce essentially endless streams of period memoranda from response streams for a similar type of order so as to the general undertaking of the model enters as a permanent position difficulties in a rather abnormal domain. Over practical limitations (similar to dealings nowhere intense over time axis) here the area of the streams limit time detail may be enter to accept two usual numerical models at philosophy of junction, which properly communicate for the normally two basic PADS algorithms. As a Banach’s fixed point result and also a Matrix junction guide to the positive departure (in fact a typical rest concept), and also junction in the logic of whole fractional commands and Kleene’s fixed point result explains the conventional variation (through its feature). In cooperation, the regular successive for a fixed point result is change instead of its “asynchronomous” or “floated” option which is explained detailed in D. Bertsekas (1989) and K. N. Chandy (1989) particularly a distributed resolution of mathematical and so many other remaining problems.

4.5.1 Split Queue Time Warp

Observing Linear Programming as routine developing tasks, we suggest calculated (planned) functions which create limited preliminary components in which their production stream into
restricted original components for their entering streams. The accurate connection among those is for output “how much” and for their input is also “how much” significant which mainly depends on characteristic for the structure in addition to know these are mentioned in the real encoding. An illustration which is the typical necessity in conventional PADS which is a algorithm, and then it’s consistent techniques, must take “look in front”: an linear programming containing study messages until time t (thus LVC is t) will in future only create information for timestamp greater than or equal to \([ t + d ]\) on behalf of positive constant d. In turn around the view, it is necessary to know that whether a linear programming has required it’s all previously consumed put in order to create its main external performance. Here if additional preceding input may be led to supplementary resulting communication or still disregarded main output. Obviously, this kind of problem is important for a hopeful method; in which rollbacks and particularly cascades for them forms a main cost hence it is neglected. In above article of Salzburg, It is considered that concluding of such kinds of difficulties then utilized them in favor of an over simplification for time warp. Generally regular time warp, in future known as a TW, linear programming is managed in order to process arriving incident messages in exacting timestamp sort. This approach is protected but may be over alert since a subsequently message can also be totally unrelated for a next progress of getting linear programming. A usual case is management of representation for elements.

Example: (See figure 4.5.1) Let taxi-cabs and customers arriving at a taxi stand. Now if a customer is already present at there, the next output of the taxi-stand linear programming mainly relates on accessibility for complimentary taxi and then a not for coming of additional clients. Accordingly, in current situation, Linear Programming possibly will neglects received communications which represent client comings along with as a replacement for look of a communication which represents a come backing taxi. (Here it is assumed that in this condition, dealing out a customer message is presently a duplication of the information for some existing data.)

Above state-dependent delay of message spending is too much helpful since in a time-warp circumstance received messages can be “poisoned”, i.e. they are away from order (because remaining Linear Programming or messages interval following) or still incorrect, which may explanation a rollback. In 1998, Split Queue Time Warp, SQTW, Hagenauer. Avoids the
unnecessary rollback since a message reaction lethargic or “by need”. At the end, a Linear Programming have individual a number of input queues, and also they have a dissimilar lines communicate in the direction of respective dissimilar communication kinds from which pending of some types communication be doable (for that refer diagram 4.5.2). Fundamental error or refusal waiting revival advance of time warp may be widespread in order to effort with new understanding for which the need for linear programming.
Rollback is exposed by equating their different for the input queues in opposition to how much distance of relative lines be previously interpreted. Thus “past” of Linear Programming is currently enclosed through sequential standards (above all gives instance for a mainly a moment ago read message of their corresponding queue), while their outlook (Here its personal messages are send) is mentioned by a highest out of these values. Otherwise, to put it in a different way, thus we comprise limited structure for asynchrony besides in the direction of asynchrony from corner to corner the Linear Programming.
Figure 4.5.2 Taxi stand example: SQTW strategy without rollback

4.5.2 Simulation System

We apply model of Split Queue Time Warp in Java. Testing indicates that Split Queue Time Warp is really decreasing so many rollbacks and hence outperform Time Warp is a proper
examples. Guaranteed from these practice, we built an absolutely new structure which permit for affecting many separate occurrence reproduction models in a similar model comprehension for reproduction representation. Obviously, Time Warp is a field of Split Queue Time Warp, and we are finish a different model planned for Split Queue Time Warp by Time Warp additionally -by merging the distinct queues interested in on simplify various things which are in the background. Here not an additional mode round, since a Split Queue Time Warp linear programming must create state-depending alternatives for the variety of input message since it needs next. Moreover, we decide to contain common sequential (result-list) simulation, but for confirmation it can be preferable in several situations for new designs, in its current condition, suggests Time Warp, Split Queue Time Warp and also sequential simulation; and remaining possibilities may be decided in future. The architecture forcefully suggests an object learning method and, mainly, drawing model (In 1994 Gamma) here we apply Java as a performance language, but perhaps will create and translation as soon as the next correction for the verbal communication enlarges OO supports (eventually offer a comparable for Java’s crossing point create). Here through structure, we assume linear programming for small granularity so that many linear programming are mapped for the different computer. Therefore, they form a group into subordinate model look for that it is suggested by a subordinate model-administrator which, as a factor, gets a policy for arrangement its linear programming. In order to assist this arrangement, our linear programming is not threads however they are parameters which recommend a one-step technique which is known as iteratively starting outer, by a sub model-executive. As observe that disconnection of simulation techniques and model description, normally we attain the sought-for reliability for the opening of two category hierarchies (In 1994; the “bridge pattern” included in Gamma) The starting hierarchy (use the word linear programming for its elements) indicates variety which give the message and simulation mechanism (here not explanatory), then the next pecking order supports in favor of a reproduction-precise details, that is explains a division for a model in consideration. Now utilize a phrase Model Entity, (ME), especially for current next group for lessons. Since explained earlier, contrasting reproduction machinery (linear programming course) have special requirements for which they must be familiar for the model entity they try: a rollback able linear programming i.e. be obliged to save model entity state for probable rollbacks. For that reason include subsequent ladder of curriculum which describe properly enhance boundary for model entity. Similar to model entities existing adapters are modeler’s
liability, which may be clear in nearby association for model entity within query. Here that’s attractive situation in highest boundary in origin group for model entity hierarchy as well as from minor changes toward techniques, however in accumulation for sustain reproduction system may disagree with this imaginary maximally. Conversely, the substance of adapters be joined in to linear programming in modify subordinate division step; however we consider that grouping comparatively a sub classing gives totally reflexivity (adapter and linear programming hierarchies is not accurately parallel) except the chance of handling unusual concerns.

Here Figure 4.5.3 as below denotes a sketch of the class diagram; where we remark only on main points.
Figure 4.5.3. Class diagram

Model entities represent conventional event-oriented simulators; in which they mention a condition and some condition exchanges procedure. These conditions are activating using the receiving of messages, and the modeler has a responsibility to apply a complete method for each kind of message which is sent in to the model entity. As regards the option for the proper procedure: we apply the “Command Pattern” as a communication forwarding, then
accomplishment is restricted in a communication which mainly hangs on mutually kind of accurate communication as well as kind of beneficiary, that is we wants to utilize twice transmit. During order to study operations, modeler be capable of apply through communication technique which present at start in theoretical representation body and also apply there by allocation for the associated linear programming (which includes a crucial performance of throw message by invoking accepted message at the end linear programming where saving a duplicate of rollbacks).

The linear programming hierarchy initiate with a thought divide into different levels to decide the variation involving obtain message, which is known as a remotely by other linear programming, and send message ; these are called only locally by the sub model administrator correspondingly the connected Model entity.

Above three procedure create the linear programming boundary (distinguish by mechanism techniques that we don’t argue here); the subsequent stage for the class hierarchy are apply only. In the Split Queue Time Warp case, the measure characteristic is a mailbox for received messages, a situation load up for a rollbacks and a productivity of buffer to keep in mind sent messages applying a same purpose. Carryout step is blocked into

a. Study messages via mailbox then arrange them into proper queues, by the help of connected adapter,
b. Test consistency which insist either rollback or a position discount,
c. Implement subsequently message (assign to model entity for choice).

Adapters have to supply linear programming in which set arriving messages into sequence (first in Time Warp, other in Split Queue Time Warp) and selecting the adjacent to-be-processed message (by reflecting in Split Queue Time Warp). They also apply the reminder model to let the linear programming restore the appropriate part of Model entities changeable situation for rollbacks.