

# Evaluation of Parallel System using Process Algebra

Ankur Mittal, Abhilash, R.P. Mahapatra

**Abstract**— In this paper we discuss method for efficiency testing of a concurrent processes execution system. We use the concept of process algebra, it is an algebraic technique for the study of execution of parallel processes. Mathematical language is use for building models of computing system which make records about the execution of the procedure. We use PEPA tool, TAPA tool for making model. These tools provide formal explanation of computing system models. The execution related data about the system will be use to check the execution efficiency of the procedure. Here we use concept of markov chain analysis for execution of the concurrent processes.

**Keywords**— Process Algebra, PEPA, TAPA, Parallel System

## I. INTRODUCTION

### A. Algebra:

It is a part of science which explores the relations and properties of numbers by methods for general images. It incorporates number hypothesis, geometry and analysis. In general structure Algebra is the investigation of scientific images and imperatives for controlling these symbols. It incorporates rudimentary condition tackling and investigation of reflections like rings, gatherings and field. The fundamental pieces of Algebra are called basic Algebra, Elementary Algebra is commonly viewed as basic for any investigation of arithmetic, science, or designing, prescription and economics. A mathematician who researches in part of science for example Variable based math is called an Algebraist [1].

### B. Process:

A process is a set of activities that interact to achieve a desire result. In computing, process is an object of a computer program that is being executed. It contains the program code and its current activity. A process may be made up of multiple threads that concurrently executed.

### C. Process Algebra:

It is study of concurrent processes. In computing, it is diverse family of related approaches for concurrent system. Process calculus gives an equipment for the extended explanation of data transfer, transmission, and harmonize among a cluster of self dependent system or jobs. Its tools are algebraical languages for the specification of processes and formulation of statement about them and verification of

these statements. In 1982, the term process Algebra was given by Bergstra & klop. Since 1984 process Algebra is used to denote an area of science. Here the term process algebra was sometimes used to refer to their own Algebraic approach for the study of concurrent processes, and sometimes to such Algebraic approaches in general [2].

The Algebraic approaches to concurrency are:

1. CCS: - Calculus of communicating system.
2. CSP: - Communicating sequential processes.
3. ACP: - Algebra of communicating processes.

## II. CALCULUS OF COMMUNICATING SYSTEM (CCS):

It is presented by Robin Milner in 1980. Its activities model indissoluble correspondence between precisely two members. It is a formal language, it include natives for depicting parallel arrangement, decision among activities and degree confinement. It evaluates the qualitative correctness of properties of a system such as deadlock

The syntax as follows:

BNF grammar is used to define the set of CCS processes:-  
 $P = \phi \mid a.E_1 \mid A \mid E_1 + E_2 \mid E_1 \mid E_2 \mid E_1[b/a] \mid E_1/a$   
P is the initial state changes to other state like  $E_1, E_2$  etc after a action has performed like 'a'.

### A. Empty process attribute:

$\phi$  (Empty process) is valid process of ccs.

### B. Choice attribute:

The process  $E_1 + E_2$  summarize as either as the process  $E_1$  occur or the process  $P_2$  occur.

### C. Action attribute:

Process  $a.E_1$  do an action  $a$  and reach to the process  $E_1$ .

### D. Renaming attribute:

$E_1[b/a]$  is the process  $E_1$  with all actions named  $a$  renamed as  $b$ .

### E. Restriction:

$E_1 \backslash a$  is the process  $E_1$  without action  $a$ .

Calculus of communicating system (CCS) is the approach to contribute to the concept of concurrent processes, it works on Algebraic approach [4]. The utility of proper learning of concurrent systems, like communication, is nowadays apparent, many format has been given as framework for concurrent system. Scientist named Milner during 80's who begins the appropriate innovation in process calculi, by the means of calculus of communicating

Revised Manuscript Received on July 18, 2019.

Ankur Mittal, Computer Science And Engineering, Meerut Institute Of Technology, Meerut, Uttar Pradesh, India. (E-mail: ankur.mittal334@gmail.com)

Abhilash, Computer Science And Engineering, Meerut Institute Of Technology, Meerut, Uttarpradesh, India

R.P. Mahapatra, Computer Science And Engineering, SRM Institute of Science and Technology, NCR, Ghaziabad, Uttar Pradesh, India

system. Milner explain that in a particular computation we have various mathematical operations and their combination as required, each and every attach with real unique concept and by combination of such operation provides universal solution. In CCS [1], the Algebra of systems/processes is based on elementary actions and on operators.

- 1)  $+$  Operator refers as alternative composition or choice.
- 2)  $\cdot$  Operator refers as sequential composition or product.
- 3)  $\parallel$  Operator refers as parallel composition or merge.

### III. BASIC PROCESS ALGEBRA

1.  $P + Q = P + Q$  D1
2.  $P + (Q + R) = (P + Q) + R$  D2
3.  $X + X = X$  D3
4.  $(P + Q) \cdot R = P \cdot R + Q \cdot R$  D4
5.  $(P \cdot Q) \cdot R = P(Q \cdot R)$  D5

We have approximation map  
 $\pi_x: A_\omega \rightarrow A_\omega$  for  $x \geq 1$  described as  
 $\pi_x(m + n) = \pi_x(m) + \pi_x(n)$   
 $\pi_x(a) = a$   
 $\pi_{x+1}(ax) = a\pi_x(x)$

Interestingly, if  
 $A_x = \pi_x(p) | p \in A$  then  $(A_x, +_x, \cdot_x)$  is the another model of basic process Algebra.

$$m +_x n = \pi_x(m \cdot n)$$

Similarly

$$m \cdot n = \pi_x(m \cdot n)$$

Projective limit is used for infinite processes called  $A^\infty$ , of the structure  $A_n$ , it means  $A^\infty$  is the set of all sequences  $p = (p_1, p_2, p_3, \dots)$  with  $p_A$  and  $p_i = \pi_i(p_i + 1)$ . These sequences are called projective sequences.

Operation  $\cdot$  and  $+$  on  $A^\infty$  are defined component wise:

$$(p + q)_x = (p)_x + (q)_x;$$

$$(p \cdot q)_x = \pi_x((p)_x \cdot (q)_x);$$

The  $x$ th component of  $p$  is  $(p)_x$ .

Thus we obtain the process Algebra  $(A^\infty, +, \cdot)$

On  $A^\infty$  a metric exists:

$$d(p, q) = 0 \text{ if } p = q;$$

$$= 2^{-x} \text{ With minimal such that } (p)_x \neq (q)_x \text{ if } p \neq q.$$

the complete metric space is  $(A^\infty, d)$ , this is the metric completion of  $(A_\omega, d)$ . The  $+$  operation and  $\cdot$  operation are  $+$  continuous. De Bakker & Zucker introduced  $(A^\infty, d)$ , in 1982[1]. In 1982, to obtain infinite processes from finite ones Milner uses charts modulo dissimulation. The equation for processes which we have described is of very basic skeleton. The zeros of the Arithmetic operation lies in arithmetically explaining futuristic mathematical operators over the present job demesne which are belong to respective corresponding composition protocol.

### IV. COMMUNICATING SEQUENTIAL PROCESSES (CSP)

In era of computer technology, a formal language is used for describing patterns which is used for communication of concurrent systems i.e. communicating sequential processes. Basically it is the part of group of mathematical theories of concurrency known as process Algebra. Tony Hoare has

described CSP in 1978. CSP is use to applied in industry for identifying and authenticate the concurrent aspects of a variety of different system, it is also use for proper simulation e commerce system .

### V. ALGEBRA OF COMMUNICATING PROCESSES (ACP)

The polynomial math of conveying procedures is use for parallel execution of simultaneous framework. Essentially it is a way to deal with thinking about simultaneous systems. It is a piece of gathering identified with numerical hypotheses of simultaneousness known as procedure calculi. Jan Bergstra and Jan willem Klop in 1982 created ACP as they were discovering arrangements of unguarded recursive conditions. The development of ACP depends on polynomial math of procedures, use to make a framework which is dynamic, summed up proverbial framework for procedure. Basically an ACP is an algebra, in the sense of universal algebra. Regular expression are use to describe the process of communication between system.

#### A. Primitives:

Atomic actions are use in ACP as its primitives. Some action has special meaning, such as the action  $\delta$ , which is used to represent deadlock, and the action  $\tau$ , which represents a silent action.

#### B. Algebraic operators:

Various action we can make by using variety of operators. Such operator can be classified by using basic process algebra, concurrency, and communication.

#### C. Choice and sequencing :

The choice operator  $(+)$  is a basic operator which provide a choice between actions, the dot  $(\cdot)$  operator is a sequencing operator, which specifies an ordering an actions.

$$(a + b) \cdot c$$

It means that first take  $a$  or  $b$  then take  $c$ .

#### D. Concurrency:

It is used to allow the description about concurrency, ACP provides merge operator. It is represented by parallel symbol

i.e.  $\parallel$ .

$$(p \cdot q) \parallel (r \cdot s)$$

may perform the action  $p, q, r, s$  in any order  $pqrs, prqs, pqs, rpqs, rpsq, rspq$ .

#### E. Communication:

Binary communications operator  $'|'$  use to represent the interaction between processes.

#### F. Abstraction:

To hide certain actions of the system, the abstraction operator  $'\tau'$  is use.

## VI. TOOLS & RESULT

### A. PEPA

It is a software for models which depends on procedure Algebra approach. Here we are demonstrating a strategy for proficiency checking and an instrument to help this methodology [7]. Performance assessment process Algebra is an Algebraic language which is utilized to assemble models of PC frameworks which take data about the exhibition of the PC framework. A technique for thinking about PEPA models continues by considering the enlistment chart procured from the model using the covered up operational semantics of the PEPA language. The determination diagram is efficiently decreased to a frame where it can be dealt with as the state progress graph of the hidden stochastic process. From this can be acquired the minute generator framework of the Markov procedure. A consistent state likelihood dissemination for the framework can at that point be gotten, on the off chance that it exists. We have executed a model device which bolsters this technique from the introductory checking of the well-formalness of the PEPA model through the production of the state progress graphs to the count of execution measures in light of the minute generator framework.

PEPA is based on communicating sequential processes, Process calculus which is related with the notion of the activities which are exponentially distributed. PEPA model is aggregation of processes which works on actions. A process may perform an action autonomously (independently) or in synchronisation with other process in the system[5]. PEPA support the following operator:

#### A. Prefix:

$$(\beta, r) \cdot E$$

It is used as atomic unit of computation of the PEPA model. It denotes a process which does an activity  $(\beta, r)$  of type  $\beta$  and simultaneously behaving as  $E$ , which is derivative of the process. The rate of activity  $r$  which taken from the domain of  $R > 0 \cup T$ . If we take rate as a positive real number then activity time duration is assumed to be drawn from an exponential distribution with mean  $1/r$  time units.  $T$  is a symbol to denotes passive synchronisation where an activity of type  $\beta$  is to be executed in synchronisation with some other process, which will calculate the overall rate of execution of the shared action.

#### B. Choice:

$$E + F$$

It indicates that a process may act as  $E$  or  $F$ . Different from ancient process Algebra in which the process choice is non-deterministic, the action of processes in PEPA is determined using stochastically. Let take  $r, s > 0$  for the choice  $(\beta, r) + (\beta, s)$  the actions  $\alpha$  and  $\beta$  are executed by  $r/(r+s)$  and  $s/(r+s)$  probabilities respectively.

#### C. Constant:

$$A = E$$

It is used to represent cyclic behaviour of the model. Let  $= (\beta, r).B$ ,  $B = (\alpha, s).A$ . Here  $A$  is a process with two derivatives which perform  $\beta$  and  $\alpha$  action permanently.

### D. Cooperation:

It represents the synchronization strategy of PEPA. The process  $E$  and  $F$  are required to synchronise over the activities types in the set  $L$ . Let's take  $(\alpha, r).(\beta, s).E \bowtie (\alpha, r).(\gamma, u).F$  is a cooperation between two processes which do a shared action of type  $\alpha$  with minimum rate of  $(r, t)$  then acting as  $(\beta, s).E \bowtie (\gamma, u).F$ . The activities  $\beta$  and  $\gamma$  are perform autonomously. In  $(\alpha, r).E \bowtie (\beta, s).F$  the process  $(\alpha).E$  does not carried out because  $\alpha$  is not available in the right hand side of cooperation. The symbol  $\bowtie$  is used to represent all shared activities between  $E$  and  $F$ .

### E. Hiding:

$E/L$  relabels the action of  $E$  with silent action  $\tau$  for all types in  $L$ . So  $((\alpha, r1)E/\alpha) \bowtie (\alpha, r2).F$  does not perform on  $\alpha$  action because left hand process performs a transition  $(\tau, r1)$  to  $E$ . All  $\alpha$  Transition performed by  $E$  are similarly hidden.

### B. TAPA

TAPA stands for (tool for the analysis of process algebra). It is a tool based on graphics. The editor of TAPA use to specifying concurrent system on terms of process algebra. We can insert terms into the system by textual representation or a graphical representation. Labelled transition system has generated during runtime environment according to the given specification. For analysis of system behaviour we use model checker and equivalence checker [2]. TAPA is a software program for supporting specification and analysis of concurrent systems via process Algebra. TAPA supports the use process Algebra for concurrent systems execution. Labelled transition system are used when systems are described as process Algebras terms. We can verify properties by checking equivalence between concrete and abstract system description and by checking temporal formulae over the obtained labelled transition system[6].

### A. PEPA PLUGINS

PEPA represents execution assessment of procedure Algebra. It is a stochastic procedure Algebra intended for displaying PC and correspondence frameworks presented by Jane hillston in the 1990s. The PEPA is an extension of traditional procedure Algebras, for example, Milner's CCS and Hoare's CSP by presenting probabilistic fanning and timing of change. Exponential conveyance is use for taking rates and PEPA models are limited state model and offer ascent to a stochastic procedure, as constant time markov process(CTMC). So PEPA is utilized for investigation of quantitative properties of models of PC and correspondence frameworks, for example, throughput, usage and reaction time just as subjective properties like opportunity from stop [6].

### B. DESCRIPTION

TAPA(TOOLS FOR ANALYSIS FOR PROCESS ALGEBRA) is a graphic based tool, developed in JAVA languages, which aims is to simplify the system and check complexity of concurrent system which is to be done by



means of process Algebra. This tools has been used for study of concurrent system of the computer science curriculum at 'university degli studi di Firenze'[3].

It contain 5 component:

- 1 Editor
- 2 Runtime Environment
- 3 Model checker
- 4 Equivalence checker
- 5 Minimizer

## A. PROBLEM:

In this paper, we are using three tools to demonstrate the same problem which will show the basic behaviour of tools.

Tools:

1. PEPA
2. TAPA

## B. Proposed Model:

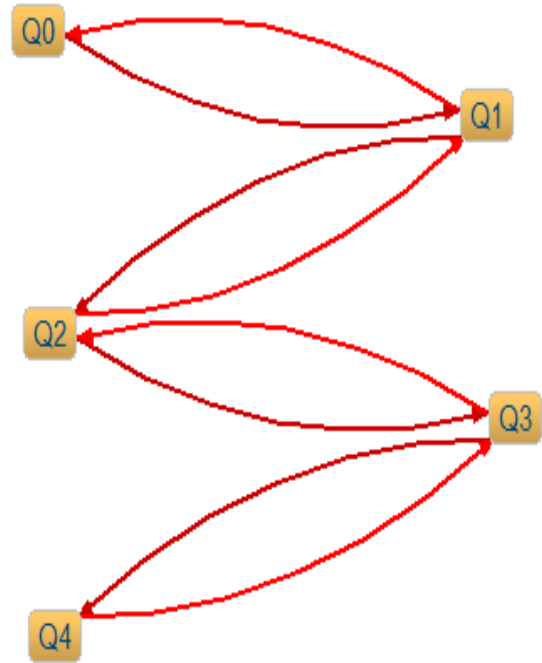
Basically in this problem we use the concept of process calculi for simulation of concurrent systems efficiently. Here a multiprocessor fault tolerance system is used, where a queue is maintained for jobs arrivals where jobs are arrived at a particular rate, we have a multiprocessor system on which we are giving a job to that processor which is free at that time. If a processor fails for any reason then it will move to fail state where First come First serve scheduling is used which decide which processor will go for repairing to the repairman. After repairing, processor is again ready for allocation of jobs and process goes on for the incoming jobs.

### A. PEPA

In this figure we have defined

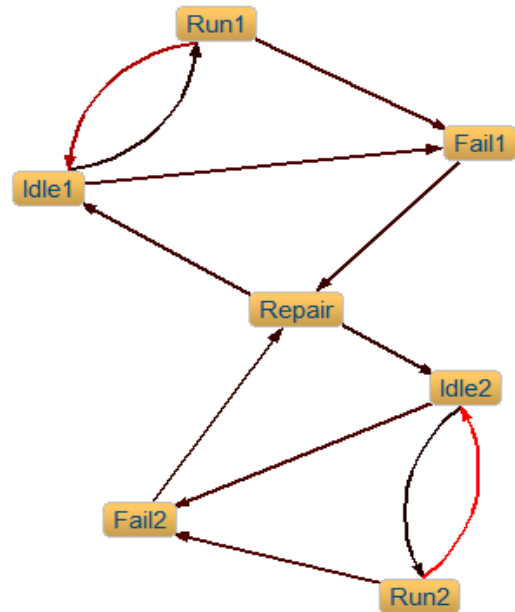
- a) The arrival rate at which the jobs comes to the queue.
- b) The departure rate at which the jobs comes to the processor.
- c) The rate at which the processor fails.
- d) The rate at which repairman takes processor repair.

$arR=0.4;$   
 $depR=0.5;$   
 $idR1=0.5;$   
 $runR1=2.3;$   
 $idR2=0.4;$   
 $runR2=3.4;$   
 $failR1=1;$   
 $failR2=1;$   
 $rep1=0.8;$   
 $rep2=0.7;$   
 $Q0=(arr,arR).Q1;$   
 $Q1=(arr,arR).Q2 + (dep,depR).Q0;$   
 $Q2=(arr,arR).Q3 + (dep,depR).Q1;$   
 $Q3=(arr,arR).Q4 + (dep,depR).Q2;$   
 $Q4=(dep,depR).Q3;$   
 $Idle1=(dep,idR1).Run1 + (fail,failR1).Fail1;$   
 $Run1=(run,runR1).Idle1 + (fail,failR1).Fail1;$   
 $Idle2=(dep,idR2).Run2 + (fail,failR2).Fail2;$   
 $Run2=(run,runR2).Idle2 + (fail,failR2).Fail2;$   
 $Fail1=(fa1,rep1).Repair;$   
 $Fail2=(fa2,rep2).Repair;$   
 $Repair=(ok1,rep1).Idle1 + (ok2,rep2).Idle2;$   
 $Q0 < dep > Idle1 < dep > Idle2$



**Fig.1 Queue Formation**

This graph shows how queue is maintain in which various jobs has come randomly and size of queue is five, when one job goes to a processor then queue pointer comes to the previous position and start pointing to the jobs available in the queue based on first come first serve strategy.



**Fig.2 Jobs Execution on Parallel Systems**

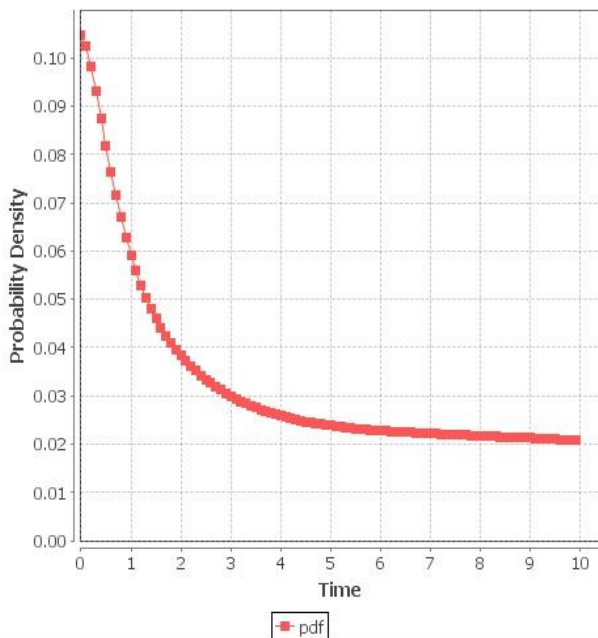




In the above figure we have three processor on which the job is allocated from the queue, when job is executing on the processor, it may be possible processor fail and it goes to repairman, after repaired it will available for assigning job, and if job is completely executed it goes to throughput.

```
Run2=(fail,failR2:1.0).Ideal1[Fail2]+(run,runR2:1.0).Ideal1
[Idle2];
Run1=(fail,failR1:1.3).Ideal1[Fail1]+(run,runR1:2.0).Ideal1
[Idle1];
Fail1 = (fa1, rep1: 1.4).Ideal1[Repair];
Idle3=(fail,failR3:1.2).Ideal1[Fail3]+(dep,idR3:0.8).Ideal1
[Run3];
Fail2 = (fa2, rep2: 1.3).Ideal1[Repair];
Repair=(ok1,rep1:1.4).Ideal1[Idle1]+(ok3,rep3:0.6).Ideal1
[Idle3]+(ok2, rep2: 1.3).Ideal1[Idle2];
Idle2=(fail,failR2:1.0).Ideal1[Fail2]+(dep,idR2:0.9).Ideal1
[Run2];
Fail3 = (fa3, rep3: 0.6).Ideal1[Repair];
Idle1=(fail,failR1:1.3).Ideal1[Fail1]+(dep,idR1:1.02).Ideal1
[Run1];
Run3=(fail,failR3:1.2).Ideal1[Fail3]+(run,runR3:0.6).Ideal1
[Idle3];
```

**Passage-Time analysis Probability Density Function**



**Fig.7 Graph shows the probability of event occurs during state change in concurrent system.**

## VII. CONCLUSION

PEPA and TAPA tool are used for making formal description for execution of concurrent processes. Both tools are efficient but PEPA is use for making mathematical description easily and give more graphical analysis about system so that we can get better result about concurrent system execution.

## REFERENCES

1. Bergstra, j. A. And klop, j. W. (1983a), "A Procedure calculi for the OperationalSemantics of Static Data Flow

Networks," Report IW222/83, Mathernatisch Centrum,Amsterdam.

2. J. Hillston. PEPA - Performance Enhanced Process Algebra. Technical report, Dept. of Computer Science, University of Edinburgh, March 1993.
3. J. Hillston. A Compositional Approach to Performance Modelling. PhD thesis, Department of Computer Science, University of Edinburgh, 1994. to appear.
4. R. Milner. Communication and Concurrency. Prentice-Hall, 1989.
5. Bradley, J., Dingle, N., Gilmore, S., Knottenbelt, W.: Derivation of passage-time densities in PEPA models using IPC: The Imperial PEPA Compiler. In: Kotsis, G. (ed.) Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, University of Central Florida, pp. 344–351.IEEE Computer Society Press, Los Alamitos (2003).
6. Robin Milner. 1980. A Calculus of Communicating Systems. Lecture Notes in Computer Science, Vol. 92. Springer.
7. Abhilash, Ram Chakka, Rama Krishna Challa, "Numerical Performance Evaluation of Heterogeneous Multi-Server Models With Breakdowns and FCFS, LCFS-PR, LCFS-NPR Repair Strategies", Advance Computing Conference (IACC), Volume :, pp. 566-570, 2013, 2013-02-22