International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Computational Smart Grid and DAM Queue

Riktesh Srivastava

***Abstract**: The layout of the Smart Computational Grid is an integral part of our information services infrastructure. In this paper, an anticipation of the architectural model of Smart Computational Grid is derived, which provides petabytes storage, tera-byte processing speed, and Gigabyte data transmission, where many supercomputers forms node with parallel data transmission. Impulsive demands posed by unexpected flows of data from users for complex computations, however, is exceedingly challenging. The paper specifies the design and implementation of a Dynamic Allocation of Memory (DAM) algorithm. DAM algorithm identifies the design of memory at supercomputing node, for ergodic functioning and for ensuring that no data overflow, using the methodology of queuing theory. The overall performance of the DAM algorithm is shown using MM1 and GG1 queue models and results are compared accordingly.*

Keywords: Smart Computational Grid, DAM algorithm, MM1 queue, MG1 queue, Ergodicity.

I. INTRODUCTION

Smart Grid computing is an extension of distributed computing where the distributed resources can be shared and used effectively [1]. Nodes and resources are dynamically added in the Computational grid [2], wherein, the requests from these nodes are forwarded to super computer for processing. These super computers have immense processing capacity due to combination of distributed computing. All the requests from nodes stay in memory and waits for its turn to be processed. It was perceived during the experimentation that the requests need to be classified based on grain size[3]. The processing of the requests need to follow parallel processing [4], where the proposed DAM (Dynamic Allocation of Memory) algorithm categorizes the requests into three classes, C1, C2 and C3, and then each queue is processed in parallel. The classification is defined as[5]:

1. Fine Grain: Request to be takes less than 5 seconds
2. Medium Grain: Request to be takes between 5 -10 seconds
3. Coarse Grain: Request to be takes more than 10 seconds

DAM algorithm is mentioned in Figure 1 mentioned below:

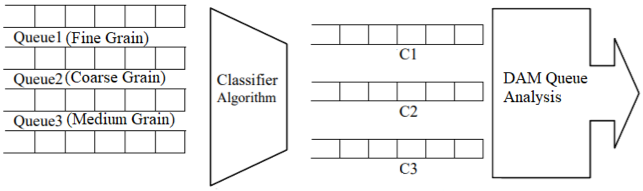

Fig 1: Layout and Interfacing of Computational Grid

We have merged all the queue classification into one queue, and conducted experimentation for DAM queue using both MM1 and MG1 queues. In the notation of queue, M is Markovian distribution and G is General distribution. For General distribution, we have choice between different types of algorithms – Bernoulli, Gaussian, Equiprobable, Negative-Exponential, Modified Geometric distribution. For the experiment, Bernoulli algorithm is taken for general distribution.

The paper is divided as follows: Section 2 gives the architecture of Smart Grid, which includes combination of super computer and nodes. Section 3 mentions the derivation of mathematical equations for DAM queue using MM1 and MG1 queue models respectively. Section 4 shows the outcomes based on mathematical derivations. Section 5 concludes the exploration with recommendations and further research.

II. ARCHITECTURE OF SMART COMPUTATIONAL GRID

In the anticipated study, the super computers form the node of Computational grid. The branches (*edge*) between two nodes are constructed using optical fibers having at least 144 cores. A dedicated cable between two nodes forms the computational grid. The layout is shown in Figure 2 below.

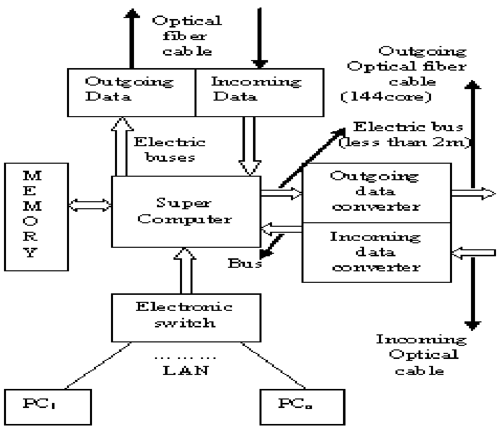

Fig 2: Layout and Interfacing of Smart Computational Grid

Manuscript published on 30 September 2019
* Correspondence Author
Riktesh Srivastava*, School of Engineering and Technology, Al Dar University College, Dubai. Email: riktesh@aldar.ac.ae

It is apparent from Figure 1, that two sets of data converters will be obligatory for the computational grid node. They should be associated to the parallel port of the super computer [6]. The electric pulses are used for Incoming and Outgoing of data to the node, wherein, these pulses are transmitted in parallel which enables the grid to make super computer to process in parallel. The biggest challenge in designing of computational grid is of memory associated with super computer [7]. With such a bursty traffic arriving at a time, the memory design should be robust to manage the outflow and inflow of data [8]. Furthermore, the size of memory should be dynamic [9], so none of the data used for computations to be lost.

III. MEMORY DESIGN FOR SMART COMPUTATIONAL GRID

Memory Design is an imperative ingredient of design of Smart Computational Grid. For λ requests arriving at the system and µ responses departing, the computational grid should presume that the functioning is stable, when $\lambda<\mu$. Such a condition is called Ergodicity [10], which defines the condition such that there is no overflow of data.

A. Derivation of Queue Length using MM1 model for DAM queue

For estimation of the Queue Length using MM1 model, following conditions are assumed:

1. Δt is value in which only one process can occur, i.e., either arrival or departure of request/response.
2. State of arrival is denoted as λ and state of departure is denoted as μ.

Probability of 1 arrival $= \lambda\Delta t$

and, the probability of 1 departure$= \mu\Delta t$

Then, probability of 0 arrival $= 1 - \lambda\Delta t$

and, the probability of 0 departure $= 1 - \mu\Delta t$

Considering n data to present at any time t and is denoted by $P_n(t)$. If the time is increased from t to $t + \Delta t$, then

$$P_n(t+\Delta t) = \begin{cases} P_n(t)(1-\lambda\Delta t)(1-\mu\Delta t) \\ P_{n+1}(t)(\mu\Delta t) \\ P_{n-1}(t)(\lambda\Delta t) \end{cases} \quad (1)$$

Upon solving eq. (1), we get

$$P_{n-1}(t)\lambda - (\lambda+\mu)P_n(t) + P_{n+1}(t)\mu = 0 \quad (2)$$

Assuming there were 0 requests at time $(t + \Delta t)$, eq. (2) can be stated as:

$$P_0(t+\Delta t) = P_0(t)(1-\lambda\Delta t)$$
$$= P_1(t)\mu\Delta t$$
$$= P_0(t)(1-\lambda\Delta t) + P_1(t)(\mu\Delta t)$$
$$\left(\frac{P_n(t+\Delta t)-P_n(t)}{\Delta t}\right) = P_n(t+\Delta t) = -P_0(t)\lambda + P_1(t)\mu$$

(3)

For stable condition, eq. (3) is set to 0, we get the following

$$P_1(t) = \left(\frac{\lambda}{\mu}\right)P_0(t) \quad (4)$$

Generalized equation becomes

$$P_n(t) = \left(\frac{\lambda}{\mu}\right)^n\left(1-\frac{\lambda}{\mu}\right) \quad (5)$$

Queue Length can be thus mentioned as

$$Q_L(MM1) = \sum_{n\to\infty}^{N} nP_n(t)$$
$$= \sum_{n\to\infty}^{N}\left(\frac{\lambda}{\mu}\right)^n\left(1-\frac{\lambda}{\mu}\right)$$
$$= \left(1-\frac{\lambda}{\mu}\right)\sum_{n\to\infty}^{N}\left(\frac{\lambda}{\mu}\right)^n$$
$$Q_L(MM1) = \frac{\left(\frac{\lambda}{\mu}\right)}{\left(1-\frac{\lambda}{\mu}\right)} \quad (6)$$

Eq. (6) can be unswervingly applied to estimate the value of queue length. Also, size of queue should also include Standard deviation for overall derivation of queue length [11], as given below:

$$\widehat{M} = Average\ Queue\ Length + Standard\ Deviation$$

It was also observed that Standard Deviation is same as Average Queue Length, thus, overall queue length is mentioned as

$$\widehat{M} = 2.\left(\frac{\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}}\right) \quad (7)$$

B. Derivation of Queue Length using MG1 model for DAM queue

It was observed during study that MG1 model follows second order polynomial equation [12] as mentioned in eq. (8) below

$$Q_L(MG1) = y_i = a_0{x_i}^2 + a_1x_i + a_2 \quad (8)$$

where a_0, a_1 and a_2 are coefficients of polynomial for MG1 model.

Eq. (8) gives the queue length for larger values of data and not for lower values.

There are two challenges while deriving the queue length for MG1 model:

1) Finding the values of coefficients a0, a1 and a2
2) Identifying the value of $x(i)$

Values of a_0, a_1 and a_2

To calculate the coefficient values, "S" represents the error in computation and can be mathematically represented as

$$S = \sum(y_i - \hat{y}_i{}^2) = \sum(y_i - a_o{x_i}^2 - a_1x_i - a_2)^2 \quad (9)$$

If be differentiate S w.r.t a_0, a_1, a_2 and setting each of these coefficients equal to zero, we get

$$\left.\begin{aligned} na_0 + a_1\sum x_i + a_2\sum x_i^2 &= \sum y_i \\ a_0\sum x_i + a_1\sum x_i^2 + a_2\sum x_i^3 &= \sum x_i y_i \\ a_0\sum x_i^2 + a_1\sum x_i^3 + a_2\sum x_i^4 &= \sum x_i^2 y_i \end{aligned}\right\} \quad (10)$$

where "n" is the degree of polynomials, which can be solved using Gauss Elimination method.

Table 1 shows the summations of variable rate of arrivals, $Q_L(MG1)$, their products and powers.

Table- I: Variables of M/G/1 model

$x_i=\lambda$ / $y_i=Q(L)$	5000	7500	10000	12500	15000	$\sum x_i = 50000$
y_i	5102	7653	10204	12755	15306	$\sum y_i = Q_i = 51020$
x_i^2	2500000	56250000	100000000	156250000	225000000	$\sum x_i^2 = 562500000$
x_i^3	125000000000	421875000000	1000000000000	1953125000000	337500000000	$\sum x_i^3 = 687500000000$
x_i^4	625000000000000	316406250000000	1000000000000000	244140625000000	506250000000000	$\sum x_i^4 = 888281250000000$
x_iy_i	25510000	57397500	102040000	159437500	229590000	$\sum x_iy_i = 573975000$
$x_i^2y_i$	127550000000	430481250000	1020400000000	1992968750000	3443850000000	$\sum x_i^2y_i = 7015250000000$

When the last column of the table is substituted in normal equation, we get the following three simultaneous equations for variables, a_0, a_1 and a_2.

Normal equations are:

Retrieval Number: C6428098319/2019©BEIESP
DOI:10.35940/ijrte.C6428.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$5a_0 + 50000a_1 + 562500000a_2 = 51020$$
$$50000a_0 + 526500000a_1 + 6875000000000a_2 = 573975000$$
$$526500000a_0 + 6875000000000a_1 + 88828125000000000a_2 = 7015250000000$$

By Gaussian elimination and using back substitution, we obtain:

$$a_0 = 1.0089 \text{ X } 10^{-7}$$
$$a_1 = 1.001$$
$$a_2 = 1.68 \text{ X } 10^{-6}$$

Value of $\boldsymbol{x(i)}$

To identify the value of $x(i)$, we use Bernoulli algorithm as mentioned below:

```
begin
read λ,n
b=1/λ
a=1-b
for i=1 to n in step of one do
q(i)=RAND U(i)
x(i)=(-a+SQRT(a*a+2*b*q(i))/b
write(x(i))
end for
end
```

IV. OUTCOMES

The outcomes of the research conducted is mentioned in Figure 3, which clearly indicates that for the given number of requests arriving, MG1 model superseded MM1 model. The result evaluates the queue length, which indirectly identifies the memory size of the supercomputing node at the Computation Grid (as mentioned in Figure 2). However, the following conditions were followed to observe the outcomes:

1. Arrival should be less than a threshold value (predecided < μ).
2. In general, if memory is designed as derived in Eqs. (7) and (8), there will be no overflow of data.
3. All the class of requests (C1, C2, and C3) were merged into one queue to get the queue length.
4. If the size of request is large, then host computer assigns the module for the program, to rest of the other supercomputers. The other supercomputers after processing the program module return the program back to the host computer, which in turn, sums up the result give the desired output.

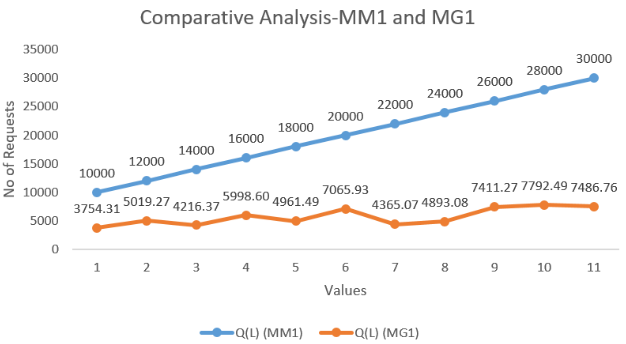

Fig 3: Outcomes based on MG1 and MM1 queue

V. CONCLUSION

Before y The outcome gives the clear indication that MG1 model should be adopted for query processing for Computational Smart Grid. As mentioned in the outcome, the queue length (or memory size) required will be almost half, if we use MG1 model, or in other words, Computational grid can be used to process complex computations fast. But, these outcomes are based on certain predefined conditions, and moreover, other queue models are also to be tested for the same number of requests. DAM queue can also use GM1 and GG1 queue models, which will be used and compared for future research.

REFERENCES

1. F. P. Guimaraes, P. Célestin, D. M. Batista, G. N. Rodrigues, and A. C. M. A. de Melo, “A Framework for Adaptive Fault-Tolerant Execution of Workflows in the Grid: Empirical and Theoretical Analysis,” *Journal of Grid Computing*, Oct. 2013.
2. R. Garg and A. K. Singh, “Adaptive workflow scheduling in grid computing based on dynamic resource availability,” *Engineering Science and Technology, an International Journal*, vol. 18, no. 2, pp. 256–269, Jun. 2015.
3. W. Jonas and V. Andreas, “Automatic Classification of Requirements Based on Convolutional Neural Networks,” in *IEEE Digital Library*, China, 2016, vol. 1, pp. 39–45.
4. Kai Hwang, “Advanced parallel processing with supercomputer architectures,” *Proceedings of the IEEE*, vol. 75, no. 10, pp. 1348–1379, Oct. 1987.
5. R. Srivastava, “Evaluating the QOS for E-Commerce Architecture Using Proposed WQM Algorithm,” in *Information Systems International Conference (ISICO), 2 – 4 December 2013*, Bali, Indonesia, 2013.
6. M. Galle, T. Gerhold, and J. Evans, “Parallel Computation of Turbulent Flows around Complex Geometries on Hybrid Grids with the DLR-TAU Code,” in *Parallel Computational Fluid Dynamics 1999*, D. Keyes, A. Ecer, J. Periaux, N. Satofuka, and P. Fox, Eds. Amsterdam: North-Holland, 2000, pp. 223–230.
7. P. Balaji, B. D, and K. D, “Scalable computing and communications: theory and practice,” in *Fault tolerance techniques for scalable computing*, Hoboken, New Jersey: John Wiley & Sons Publishing, 2012.
8. R. Oldfield and D. Kotz, “Improving Data Access for Computational Grid Applications,” *Cluster Comput*, vol. 9, no. 1, pp. 79–99, Jan. 2006.
9. I. Foster and C. Kesselman, Eds., *The Grid: Blueprint for a New Computing Infrastructure*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
10. J. D. Cordeiro and J. P. Kharoufeh, “The Unreliable M/M/1 Retrial Queue in a Random Environment,” *Stochastic Models*, vol. 28, no. 1, pp. 29–48, Jan. 2012.
11. D. L. K. Singh and R. Srivastava, “Estimation of Buffer Size of Internet Gateway Server via G/M/1 Queuing Model,” *International Journal of Computer and Information Engineering*, vol. 1, no. 9, p. 9, 2007.
12. L. K. Singh and R. Srivastava, “Memory Estimation of Internet Server Using Queuing Theory: Comparative Study between M/G/1, G/M/1 & G/G/1 Queuing Model,” *International Journal of Mathematical and Computational Sciences*, vol. 1, no. 9, p. 5, 2007.

Retrieval Number: C6428098319/2019©BEIESP
DOI:10.35940/ijrte.C6428.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication