ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

Speed up Optimization in Distributed Systems

Dr. P. Bayat[1], L. Omrani[2], S. Bayat[3]

Researcher, Mathematics and Informatics Research Group, ACECR, Tarbiat Modares University, P.O. Box: 14115-343, Tehran, Iran[1]

Lecturer, Dept. of Computer Engineering, Payame Noor University, Gilan, Iran[2]

Msc. Student, Dept. of Computer, IAU of Arak, Arak, Iran[3]

Abstract: This paper presented a new approach to simulation of distributed systems. There are some limited approaches to simulation of distributed systems, such as petri-net, mathematical simulations and some softwares like NS and etc. In Internet as a semi-distributed system, the methods for preventing limitations are very important. Therefore, in this research a new approach for simulating distributed systems is introduced to expand the abilities and decrease the limitations of expanding and maintaining the system bottlenecks. The new idea is based on simulation of mechanical elements by some of electrical elements and obtain some formulas as the result of this simulation. In the new method, after understanding the relationships between parts of a distributed system and electric elements (like the similarity of these elements and mechanical behaviours), simulation will be begun and it uses the rules which are proved in electric lows to optimize these systems. Because in order to analyse these systems with any kind of size, it can uses some kinds of software that are used in electrical science a lot and it can solves some basically problem of distributed systems that the past approaches unable to solve them, such as incremental growth and performance (speed) optimization. In this research, both of static and dynamic behaviour of systems are important.

Keywords: Distributed systems, simulation, electric elements, optimization

I. INTRODUCTION

Distributed systems have the most performance networks. Performance optimization in these systems can help users being to get more processing power and save the time. In the context of hardware and software systems, formal verification is the act of proving or disproving a priority of a system with respect to a formal specification, using methods rooted in mathematics, such as logic and graph theory [9]. A formal specification of a system can help to obtain not only a better (more modular) description, but also a better understanding and a more abstract view of the system. Formal verification that supported with automate tools, can detects errors in the design that are not easily found by using related tests, and can be used to establish the correctness of the design. Formal verification has been applied to communication and cryptographic protocols, distributed algorithms, combination circuits, and software expressed as source codes. A comprehensive of the field of formal verification can be found in [7]. Process characteristics focuses on the specification and manipulation of process terms that inducted by a collection of operator symbols. Such a process term constitute a formal specification of a system. Typically, process characteristics contain action names, to express atomic events, and two basic operators consist of: alternative and sequential composition to build finite processes. Recursion allows one to capture infinite behaviour. Verifying the correctness of distributed systems is a challenge, due to their inherent parallelism. In order to know, the behaviour of distributed systems in detail, it is imperative that they are dissected into their concurrent components [7]. Generally, a system is consists of a mix of processes and data [7]. Processes are the control mechanisms for the manipulation of data. While processes in the system are dynamic and active, the system's data are static and passive [10]. It means processes (requests) are changeable during the time and data paths (flows) are constant. In this research, the function of distributed systems is discussed and it introduced a kind of simulation to optimize the function of distributed systems. For this purpose both of the hardware and software of these systems are discussed and tried to link these two items to electric elements. In the next section, after some definitions and issues, the new simulation present and in section 3, operations and their behaviours will be explained. In section 4, will discuses about power and energy in electronic circuits, and in section 5, analysis the new simulation and finally in sections 6, 7 conclusion of analysis simulation and future works about this new approach considered and presented.

II. SIMULATING TOPOLOGIES OF DISTRIBUTED SYSTEMS

II.I. DEFINITIONS

II.I.I. HARDWARE

In a distributed hardware there is a loose connection among computers [5],[6] and [7] so, there is Seri of parallel impedances. Of course, multiprocessors tend to be more tightly coupled than multicomputrs, because they can exchange data at memory speeds, but some fibber optic based multiprocessors can also work at memory speeds. Normally, hardware behaviour in distributed systems is as parallel impedances. A single computer model can substitutes as each of impedances, now each box of impedances considers. Basically, each of computers in a network is includes some main part such as CPU, MEMORY, MODEM (or network card) and also HUB. The reason that hub is mentioned in this list is the security for the related information and data [2], [3], [4].

Assume that the current of electrons is similar to the current of packets (or every type of data such as frames, bits and etc.) that moves in the network, and the voltage is similar to the resources of network (including software and hardware resources) such as a place for data storage. Therefore, the simulation of electric elements performs as follow:

- Memory has the concept of storing data, and capacitor has also the concept of storing voltage. Clearly, states are considered for a capacitor; primary or static state ($V0(t)$), and dynamic state ($v(t)$). In this model the primary or static state of capacitor as a hard disk of computer, and dynamic state as temporary memory in the computer are considered.
- Modem and communication lines are causes of the delay in sending and receiving data. Another effect of any communication line is creating a connection between a computer and others. In the electric circuits, a resistance is an element that resists against the movement of electrons, an agent in the networks to suppress the movement of packets, is delay in communication lines. So these two subjects and behaviours are similar.
- Hub has the concept of giving permission to data to enter inside the system from outside and vies versa, in the other word, it checks whether data can dart or not, so there is a simulation as which is shown in Fig. 1.
- CPU is including a cache and the related registers. Obviously, the activation of these elements is saving and gathering of data. On the other hand, a capacitor has a behaviour of saving and gathering the electrons in a circuit. Additionally, the following circuits are correspond to the computer elements:

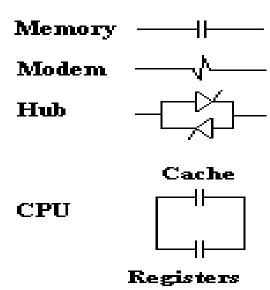

Fig. 1 Memory, modem and CPU model

Its should also be mentioned that hub can have a very important role to control the movement of data. The word "security" in electronic engineering has a close relation to the circuit protection with a diode element. In a computer network a hub can use to prevent hackers to enter our system (site). Installing of two diodes (in the ends of circuit), is used to more control and security operations (input and output controls). So the suggested circuit for each Impedance Zi can be as the following:

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

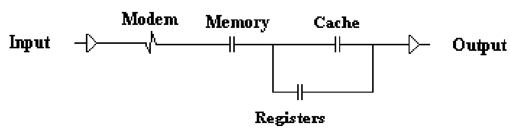

Fig.2 Data flow in simulated computer

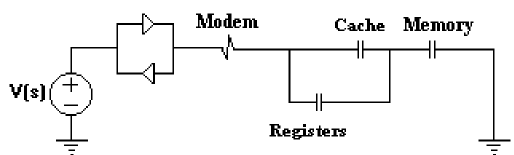

Fig.3 A represent model for a computer

Regarding to hardware aspect, as mentioned, in this work, focus is on more speedup of hardware and as the future works, have an optimized load balancing, incremental growth, system scalability that are the main problems of distributed systems. By proportioning after analysing the system by known software “pspice”, also the other problems of distributed system can be find out, when there are a large number of users. But the following pattern is able to achieve the interesting result.

In the above figure, there is a memory (or capacitor) that includes RAM, and H.D.D in temporary and static conditions. Sticking point is that Registers are similar to H.D.D. and CPU caches, and they are similar to RAM, when a computer restarts. Data in registers are constant but in this case, data in cache are dynamic. On the other hand, data negotiates sequentially for communication (send and receive) links and devices and then CPU and finally memories. Likewise, the Internet (or a distributed system) resources are modelled by the power supply, and the power supplies are in two kinds: voltage generators and current generators.

The voltage generators, stores energy (corresponding to data) and the current generators injects energy or electrons (corresponding to data) to circuits (or distributed systems). As an example, a workstation that is connected to Internet (or a distributed system) is shown in Fig. 3. The mentioned parameter (**ω**) is the rate of transferring data or data flow (frequency), and because the capacity of memory (H.D.D. and RAM) is very larger than aggregate capacity of registers and caches, due to the electric circuits lows, a simplicity can be happens (the capacity of the network is equal to the smallest capacity value). In the following figure, resistance is used instead of the collection of Modem and link line to other computer(s). Because it has delay and it causes to less power of data rate. If the link is fast and the modem has higher speed, then the corresponded resistance will have less value and in opposite, if link has a long delay and its modem has lower speed then the value of resistance will have higher value. So, with optimization and ignoring mentioned models, the above comprehensive model will be obtained. Finally, after simplifying the simulated circuit, Fig. 4 will be achieved:

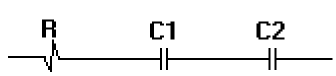

Fig. 4 Streamlined model

In this figure C1, is the summation capacity being in CPU (including a cache and registers), and C2 is the summation capacity being in memories. In a loosely coupled distributed system, according to its definition [7], the relationship between computers are loos and eliminating of a computer has not any effect others and it does not causes to crash whole of the system. This concept, notes to parallelism in electric elements, because when an electrical element is deleted, the circuit will alive and continues its tasks. Namely all impedances (corresponding to the workstations) are independent. Fig. 5 phrases this concept:

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

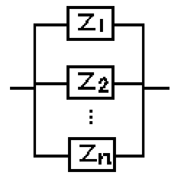

Fig. 5 Simulation of loosely coupled

If each of impedances establish by it's substituted approximate circuits, then Fig. 6 will obtain as a circuit corresponding to the considered network. **R**'s value is depend on the network delay inherently or speed of modem or network card and the **C1** and **C2**'s values are based on the related computer's memories in the network.

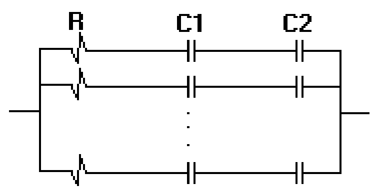

Fig. 6 Simulation of all distributed system hardware

If assume all the impedances **Z1** to **Zn** are same, then a general impedance (**Zi**) is equal to:

$$Zi = R + (1/j\omega C1) + (1/j\omega C2) \text{ -------------------- } (1)$$

Where, **C2** is upon H.D.D. and RAM. The value of **C2** is very larger than **C1**, since it includes the related registers and the CPU cache. Thus, based on circuits formula, **C2** has not a considerable effect in the result and it can aside ahead the **C1**. Therefore, **Zi** closely is equal to:

$$Zi = R + (1/j\omega C1) \text{ -------------------- } (2)$$

So, it can observe that a distributed system speed in the hardware view, is only relates to CPU storage and delay of communication lines. Finally, if **n** is number of parallel impedances, after impedance substitution (**Z**) is equal to:

$$1/Z = (1/Z1 + 1/Z1 + 1/Z1 + \dots) \text{ --------------------- } (3)$$

$$\rightarrow 1/Z = n/Z1 \text{ --------------------- } (4)$$

$$\rightarrow Z = Z1/n \text{ -------------------- } (5)$$

Where, **n** is related to the number of modelled computers. If an impedance of a circuit is lower, it has a better situation than related impedance is a large number. On the other hand, each **Z1** has an inverse relation to the **C**, so, if **C** increases then **Z1** decreases and thus, **Z** decreases (because of their direct relation). As an important result, it is better that the values of the registers and cache are upper, to achieve a more network (distributed system) speed. Accordingly a distributed system is a collection circuit of parallel resistance and capacitor that stick serial. Thus, an important factor for a circuit is the summation capacity within it's CPUs (cache). It is important that, the caches are the preponderant elements of a distributed system. For example the system can analyses with speedup or performance or etc. On the other hand, by combining two above relations this formula is calculated:

$$Z = (R1/n) + [1/(jn\omega C1)] \text{ -------------------- } (6)$$

So, when the number of impedances increases, **R1** will be less than real rate, and also the impedance is related to **j**.
Still, this circuit can analyses exactly by tools of circuit simulating (pspice*)*.

II.I.II SOFTWARE SIMULATION

Opposite the loosely coupled behaviour of hardware of a distributed system, in a software part, there is a tightly coupled behaviour. Tightly coupled systems tend to be used as parallel systems (working on a single problem) and loosely coupled ones tend to be used as distributed systems (working on unrelated problems), although most of software which uses in distributed systems are in parallel situation because one of the importance aspects of distributed systems is parallelization.
Therefore, assume a number of Multiplexers/D-Multiplexers couples are exist in a circuit as figure and a user (process) can has access to resources with selecting and enabling special MUX (DMUX), which are suitable elements and can use to

ISSN: 2319-8753

International Journal of Innovative Research in Science,

Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

resources. After determining the path of data flow, some of MUXs (DMUXs) can chosen that are included the related path. Moreover, enabling of the mentioned elements shows the data flow's paths. Off curse, when a MUX (DMUX) chooses for data path, a line of inputs and transfer it to output should select. In addition, when a user (process) wants has access to a resource, data flow will be from right to left and MUXs are active and when resources send data (reply), DMUXs will be active.

Therefore, the choosing path that enables a set of MUX (DMUX) is performs by a routing algorithm. Clearly, routers determine the best path of data flow in networks and this operation is based on routing algorithms. In the other hand, this model is conforming to the software defining in distributed systems. Because as mentioned, software in distributed systems has tightly coupled connection and if each of MUXs (DMUXs) involved to data flow down, then software operation will destroy and the connection of user (process) to resources will be disconnected.

Fig. 7 shows a simple set of MUX/DMUXs, and as explained, on the left there are 8 resources and on the right of figure there is a user or process. Of course, if the number of users or processes were more than one, they will put in right site in different times (as a real dynamic structure). In other word, for different users (processes), in different times, users (processes) should put in the right side of the bellow figure, one-by-one.

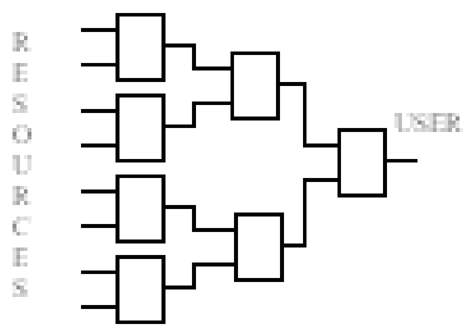

Fig. 7 The coupled of MUX/DMUXs

As shown in the above figure, for each process there are two paths at the first Multiplexer (up and down), then each of the paths has a user process that can find the best path to a resource access.

Fig. 7 is a very simple model and if exist a complex path, the MUX/DMUXs can use with different states. For example MUX/DMUXs 1x4, 1x8, etc. Electrons that are moving in the circuit can simulate with processes and data flow in distributed system. In addition, each of MUX/DMUXs, are like a module of software.

Resources can assume hardware or software because hardware resources are like software resources in software view. In the other word, if a software model considers as a client/server model in real, the user or process (left side) is client side and resources (right side) are as a server side. After configuration of software network, its inputs and outputs analysis is accessible with a simulator circuit. In this research, "pspice" is used.

III. CAPACITOR AND ITS AFFINITY BY DATA SAVING

According to the circuits laws, for each capacitor in dynamic behaviour:

$$I = C(dv/dt) \text{ -------------------- } (7)$$

$$\Rightarrow \int dv = (1/C) \int I*dt \text{ -------------------- } (8)$$

$$\Rightarrow C*V = \int I*dt = I \int dt = I*T(total) \text{ -------------------- } (9)$$

$$\Rightarrow T(total) = (C*V) / I \text{ -------------------- } (10)$$

On the other hand:

$$I*dt = C*dv \text{ --------------------- } (11)$$

$$\Rightarrow \int dx = (1/C)*\int I*d\tau \text{ --------------------- } (12)$$

$$\Rightarrow \{dx \text{ from } v(t0) \text{ to } v(t) \text{ \& } d\tau \text{ from } t0 \text{ to } t\} \text{ -------------------- } (13)$$

$$\Rightarrow V(t) = (1/C)* \int i*d\tau + V(t0) \text{ --------------------- } (14)$$

Because t0 is equal to 0 then dτ is from 0 to t:

$$V(t) = (1/C)*\int i*d\tau + V(0) \text{ -------------------- } (15)$$

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

$$\Rightarrow V(t) = (1/C)*\int i*d\tau + V(0) \quad (16)$$

In this formula the part that included integral is a dynamic behaviour of the capacitor and this is similar to RAM elements and the constant part (steady state) is similar to H.D.D.

IV. POWER AND ENERGY

Another simulation aspect is energy simulation and evaluation. There is a relationship between electric engineering and data storage in distributed systems, in energy view. At the electric devices, energy can be store in capacitors. This fact is exhibited in the below formula (for power):

$$P = V*I = C*V*(dv/dt) \quad (17)$$

$$\Rightarrow P = I*[(1/C)\int i*d\tau + V(0)] \quad (18)$$

$$\{\tau \text{ is from 0 to t}\}$$

For energy parameter:

$$dw = C*V*dv \Rightarrow \int dx = C\int y*dy \quad (19)$$

$$\{x \text{ from 0 to W and y from 0 to V}\} \quad (20)$$

$$\Rightarrow W = (1/2)*C*V^2 \quad (21)$$

$$\Rightarrow W \propto C \quad (22)$$

So, the rule of **C** (capacity of energy or data) and **W** (energy or accumulation of data) are commensurate. On the other hand, the concept of vectorial multiply defines at databases [9].

V. ANALYSIS

This section presents the plots of a distributed system which is simulated by "OPNET" software. The measured speed up have shown in first part of Fig. 8. On the other hand, after analysis of the simulated circuits (corresponding to the networks), the second part of Fig.8 achieved that analysed by "pspice".

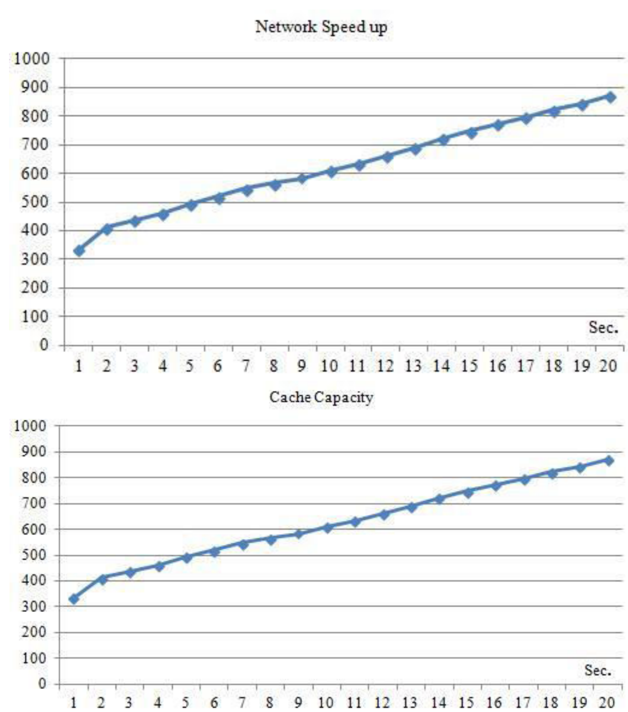

Fig. 8 the relationship between a network speed up and cache capacity in its corresponded simulation

There is an important point that the networks features were same and only their equivalent caches were different.

In spite of having many new software that can be very helpful to simulate distributed systems, this method needs to have a right order, the point in this method is very trustable because it is based on electric rules and the process of simulation can be physically felt [8].

As another advantage of the presented simulation, can mention to the speed of simulating process that is very high.

VI. CONCLUSION

In this paper, a new method for simulating of distributed systems based on electric elements is introduced. Also, the simulating of both software of distributed systems and the related hardware is discussed [7].

By discussing this issue a good result that stems in electric rules can achieve, that is proved in mathematic and can be trustable. So, by this optimization more acceptable results can accomplish.

VII. FUTURE WORKS

The mentioned discusses was the beginning of a great simulation of distributed systems that was introduced as a hypothesis. It must be mentioned that in this research, important topics like security, software design [7], Internet engineering and also other topics that are related, were given a good reason for having research about.

Furthermore, the conclusion to optimize the security, load balancing and also communication of the system are used [1]. As the first step, the relation between hardware and software are proved to have strong proof for what is concluded.

At the end, authors of this paper are eagerly ready to hear any related suggestion of respectable readers.

REFERENCES

[1] P. Bayat, and M. Challenger, "A New Reliable Distributed Mutual Exclusion Algorithm" 9 th IASTED conference of Software Engineering and Applications, Phoenix, USA, 2005.

[2] Bayat, P. and Challenger, M., "Simulating of distributed systems by electric elements", 3rd conference of electrical engineering in Ardebil-Iran, 2003.

[3] Bayat, P., Challenger, M., "A new method in simulating of distributed systems", 2nd Iranian National conference of computer in Mashhad-Iran, 2004.

[4] P.Bayat and M.Challenger, "An optimization in distributed system" 3rd conference of electric and computer engineering in Mazandaran university-Iran 2004.

[5] Tanenbaum, A.S.: "Operating Systems, Design and Implementation," Prentice-Hall Int., Inc, 1997.

[6] A.S.Tanenbaum, "Distributed Operating Systems," Prentice-Hall International, Inc, 1995.

[7] A.S.Tanenbaum and M.V.Steen "Distributed Systems Principles and Paradigms", Prentice-Hall International, Inc, 2008.

[8] A.J. van der Schaft, "Port-Hamiltonian systems: a theory for modeling, simulation and control of complex physical systems", Ph.D. thesis, 2005.

[9] A. Grimaldy, "Discret Structures", Prentice-Hall Int., Inc, 2003.

[10] Cali F., Conti M., Gregori E., Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical limit. IEEE/ACM Transactions on Networks. 8(6): 785-799, 2000.