

Cloud Computing and Robotics for Disaster Management

Nitesh Jangid

Information Technology Department
Green Research IT Solutions Pvt. Ltd.
Jaipur-302023, Rajasthan, India
Email: nitesh_jangid@greenresearchit.com

Bindu Sharma

Amity School of Fashion Technology
Amity University Jaipur
Jaipur-303002, Rajasthan, India
Email: bsharma1@jpr.amity.edu

Abstract — Nowadays, IT community is experiencing great shift in computing and information storage infrastructures by using powerful, flexible and reliable alternative of cloud computing. The power of cloud computing may also be realized for mankind if some dedicated disaster management clouds will be developed at various countries cooperating each other on some common standards. The experimentation and deployment of cloud computing by governments of various countries for mankind may be the justified use of IT at social level. It is possible to realize a real-time disaster management cloud where applications in cloud will respond within a specified time frame. If a Real-Time Cloud (RTC) is available then for intelligent machines like robots the complex processing may be done on RTC via request and response model. The complex processing is more desirable as level of intelligence increases in robots towards humans even more. Therefore, it may be possible to manage disaster sites more efficiently with more intelligent cloud robots without great loss of human lives waiting for various assistance at disaster site. Real-time garbage collector, real-time specification for Java, multicore CPU architecture with network-on-chip, parallel algorithms, distributed algorithms, high performance database systems, high performance web servers and gigabit networking can be used to develop real-time applications in cloud.

Keywords – Real-Time Cloud (RTC); Cloud Robotics; Real-Time Garbage Collector; Real-Time Specification for Java (RTSJ); Network on Chip(NoC)

I. INTRODUCTION

The Cloud Computing is a large-scale distributed computing architecture which delivers storage and computing services to the organizations and individuals over the Internet. In fact, Cloud Computing provides a web abstraction layer above grid computing technology of supercomputing. Therefore, Cloud Computing is evolved out of grid computing and requires it for infrastructure support. There are three dominating factors contributing to Cloud Computing[1]:

- Economical multi-core architectures
- Exponential growth of data size
- Wide-spread adoption of web applications

Nowadays, robots do various tasks in many fields like industries, domestic, services, entertainment and space. Some interesting research is going on flying robots as well. Humanoid robotics is an emerging and challenging research field, which has received significant attention during the past

years. Even, Robots are not limited by onboard computation for complex decision making as in 2010, James Kuffner at Google introduced the term “Cloud Robotic”[2]. Real Time Cloud Computing[3] possibilities are also suggested by taking care of performance in hardware, operating system, web server and networking in cloud. A survey of research on cloud robotics and automation has reflected the scientific community involvement on cloud robot R&D[7].

If cloud computing facilitate real-time responses to robots then the cloud robotics may be the great way to handle disaster site by injecting various functions there. Cloud robotics will decrease the cost of robots with respect of their intelligence. If some RTC will be developed across countries by adopting some common standard then it may become the best way to handle disaster site quickly by man and machine. We can assume that a hospital at disaster site having cloud surgical robot may get help on complex diagnosis from cloud.

Java is a popular framework to develop the applications for the cloud where Java Virtual Machine (JVM) is optimizing runtime performance with overhead of Garbage Collector (GC). GC having unpredictable pauses in the application runtime and interrupting java for real-time application development. But, Real-Time Specification for Java (RTSJ)[11] standards allow developers to develop real-time application with Java by changing existing programming model. Also, incremental approach of garbage collection like used by Metronome GC[10] can decrease the application pause time with respect to non real-time GC.

The evolution of deep-submicron technology is enabling the development of multiprocessor systems on a chip(MPSoCs). System-on-Chip(SoC) opens up the feasibility of a wide range of real-time applications needing massive parallel processing. Network-on-Chip(NoC) platforms have been emerged as a scalable solution to address communication demand in MPSoCs[5][6]. Hardware parallelism is exploited by operating system, database system, web server and web applications to increase performance. Several areas of task mapping, chip virtualization and fault tolerance are under research in NoC based MPSoCs. MPSoCs are now common in cloud servers and embedded systems due to good profile on power consumption and performance.

Our target here is to explain the need and possibility of cloud computing for mission critical applications and at the same time reducing the cost of more intelligent robots by taking storage and compute power of cloud computing. For example a cloud robot can identify humans at disaster site more easily based on biometric identity (fingerprint, face and iris, etc.) and assist in various activities from humans positioning to medical treatment.

This paper is organized in many sections like real-time systems classification, hardware & software performance in cloud, dynamic memory management concept, non-real-time garbage collection, real-time garbage collection, area of interest in real-time cloud, an example of human identification by cloud robot and hierarchical intelligent model for cloud robot.

II. REAL-TIME SYSTEMS

Based on criticality of timing constraints or deadlines, the real-time systems are classified as hard, soft and firm. Hard real-time system have stringent timing requirements and failure to meet deadline result in serious consequences, example-failure of power plant control system. Soft real-time have not stringent deadline and usefulness of results decreases slowly with respect to time as deadline expires, example-stock market updates.[9].

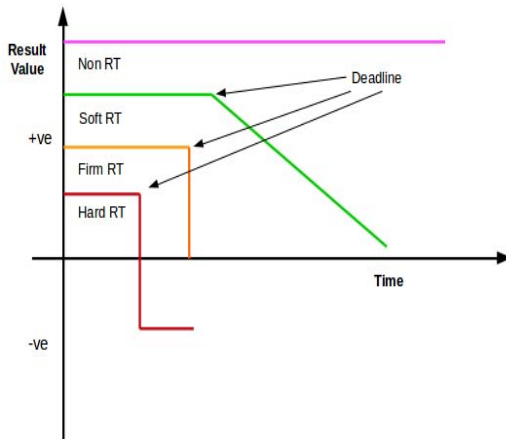


Figure 1. Real-Time System Classification

III. REAL-TIME PERFORMANCE DELIVERY IN CLOUD

Cloud Computing employs clustering to achieve high speed networking and computation. Many core processors are used to achieve high performance. Now, complex MP-SoCs are used in cloud to achieve high throughput as well as. Operating System of cloud employ best parallel opportunity of hardware by task partitioning on multi cores for applications specially designed with parallel algorithms. Web Servers based of Java technology are giving good performance by employing parallelism. Multithreading used by web applications are common in Java applications for performance. Community networking going to be better now

in terms of speed. Target clients are using more powerful computing hardware and networking than decade before.

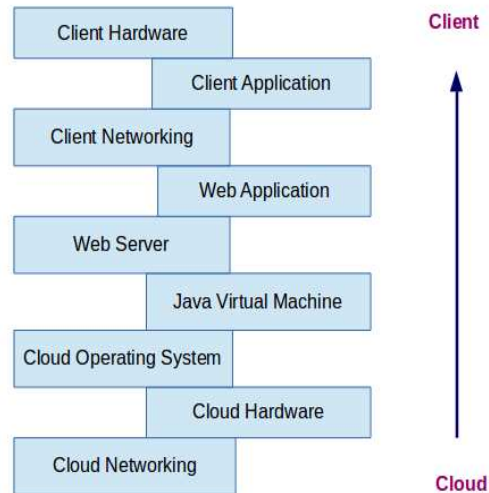


Figure 2. Software and Hardware in Cloud Infrastructure

IV. DYNAMIC MEMORY MANAGEMENT IN JAVA

As we know that the most powerful model of application development is Object Oriented Programming Model to manage complex projects. A program running into the system as a process having storage for code, global data, stack and heap. The heap is responsible to dynamically allocation of memory during process runtime[8].

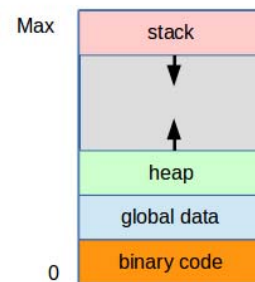


Figure 3. Process Storage Segments in RAM

In C++, If a programmer is forget to reclaim object storage from heap using delete operator then program is no longer to use object storage during its whole execution. This side effect is called Memory Leak in programming literature. One solution to above problem is garbage collector software which reclaims memory automatically as it find that particular objects are no longer in use. JVM and .Net Frameworks have their integrated garbage collector. Garbage Collector is an overhead for program execution but it ensures that there is no memory leak.

V. NON-REAL-TIME GARBAGE COLLECTOR

JVM is blessed by garbage collector which reclaim threads unused object memory from heap automatically. GC run in Java pauses running application i.e. stops application services. GC kick is generated when application threads have produced garbage on heap above some threshold. Therefore, duration and instance of GC kick cannot be deterministic i.e. it is not suitable for real-time application development[10].

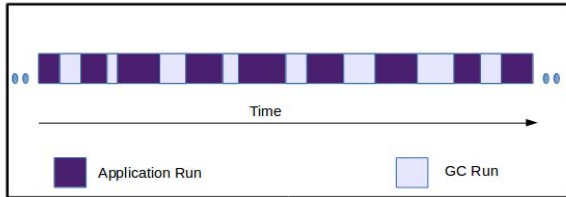


Figure 4. Non Real-Time GC Activity

VI. REAL-TIME GARBAGE COLLECTOR:

Some work is done on real-time GC by making collection activity in small increments in comparison to large quanta to maintain application predictability high. Metronome GC developed by IBM is based on this approach. GC activity can be executed in parallel to application real-time threads in RTSJ. The GC activity are more challenging and under research for multi core system as many threads running in application can introduce frequent GC run[10].

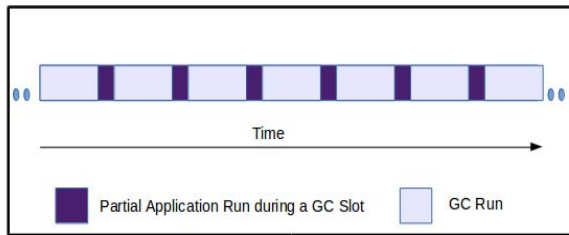


Figure 5. Real-Time GC Activity

VII. REAL-TIME SPECIFICATION FOR JAVA(RTSJ)

The RTSJ allows development of real-time application by implementing several classes for real-time thread, no heap memory, real-time clock, schedulable objects, scheduling, schedulability analysis, synchronization, asynchronous events, physical memory access, resource sharing, etc. RTSJ thread may get 28 priority level and having ability to pre-empt garbage collector thread due to higher priority. Object in RTSJ can get their storage in immortal memory(no heap memory) which is not garbage collector controlled for reclaim. Various real-time algorithms can be easily implemented by RTSJ but multiprocessor real time algorithm support is under research to get multi core advantages[11].

VIII. EVOLUTION IN MULTICORE CPU ARCHITECTURE

Chip design has important aspects like computation, memory, communication, and I/O[4]. The evolution of deep-submicron technology is enabling the development of multiprocessor systems on a chip(MPSoCs). SoC opens up the feasibility of a wide range of real-time applications needing massive parallel processing. Network-on-Chip(NoC) platforms have been emerged as a scalable solution to address communication demand in MPSoCs. A homogeneous MP-SoC simplify task migration due to identical PEs(Processing Elements) whereas heterogeneous MPSoCs support a wider variety of applications because they integrate distinct PEs like DSP, ASIC and GPP[5][6]. Several areas of task mapping, chip virtualization and fault tolerance are under research in NoC based MPSoCs. MPSoCs are now common in cloud servers and embedded systems due to good profile on power consumption and performance. Network adapter is an interface by which cores connect to the NoC and its decouples computation from communication. Routing node routing the data according to chosen protocols. Link connect routing nodes and provide raw bandwidth.

Configuring many multicore CPUs are now common in super computers, for example the top ranked (November 2015) super computer under top 500 list is Tianhe-2 which has 3,120,000 cores of Intel Xeon E5-2692v2 each operating at 2.2 GHz speed[12].

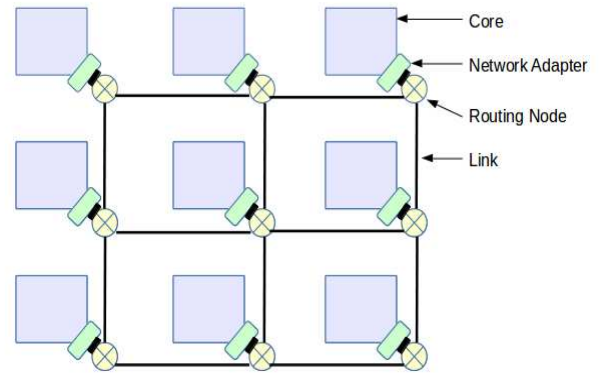


Figure 6. NoC Fundamental Components

IX. RESOURCES OF REAL-TIME CLOUD

Real-Time Cloud is possible by using appropriate multiprocessor algorithms of real-time system. Also, algorithms running on any software framework should be aware of real-time demand and need to integrate appropriate tuning mechanism or technology. Operating System of cloud should be designed by taking care of real-time performance for disaster control. There should be common standard to support cloud robotics across the countries. The important challenge is network speed and networking for cloud robotics environment

to perform at real-time demand.

X. HUMAN IDENTIFICATION BY CLOUD ROBOT

Cloud robots can identify a person at disaster site more easily based on biometric identity (fingerprint, face and iris). A city population is a big data and cannot be embedded into a robot to identify a person as it requires complex storage and processing need in a robot and increases its cost as well as. Cloud computing is equipped with powerful resource for storage and computation. Cloud can handle big data processing in real-time due to ample of resources. A cloud robot can take the biometric identification of human at disaster site and update human status and location in cloud. Moreover, It can retrieve person medical history based on its biometric identity to assist doctors in fast recovery of patient at disaster site. The discussed mechanism can work only if the valid personal and medical history records are maintained by the city government.

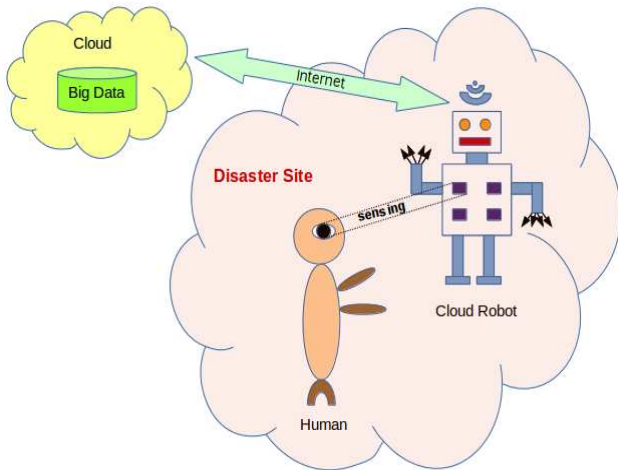


Figure 7. Human Identification By Cloud Robot

XI. DISASTER CONTROL INFRASTRUCTURE WITH CLOUD ROBOTS

Disaster site can be controlled by cloud robots using hierarchical intelligence mechanism. The slave robot is equipped with minimum intelligence to work. If next level of intelligence is required then more powerful master robot can help them. In case of master robot connectivity failure then nearby slave robots may form network and help each other as peer to peer computation. Complex processing like big data analytics, image vision, clustering, classification can be performed on Real-Time Cloud via master robot. Disaster intensity cannot be predicted with certainty therefore outer countries Real-Time Cloud may provide its services for better load sharing.

XII. CONCLUSIONS

If multicore CPU parallelism is exploited by operating system, web servers, web applications, database systems

then high performance is achieved. Also, gigabit networking (Infiniband FDR, TH Express-2, Cray Gemini and Aries) will improve cluster performance significantly. High speed of wired/wireless networking at disaster site can assist in accessing services of Real-Time Cloud. If some dedicated disaster management Real-Time Clouds will be developed across many countries then it may be possible to manage disaster sites more efficiently with more intelligent cloud robots than ever before, help hospitals to cure their critical patients within specified time frame, etc. Therefore, development of real-time garbage collection algorithms for multi core systems, real-time software technologies and fast wired/wireless Internet connectivity may play important role for mankind via cloud.

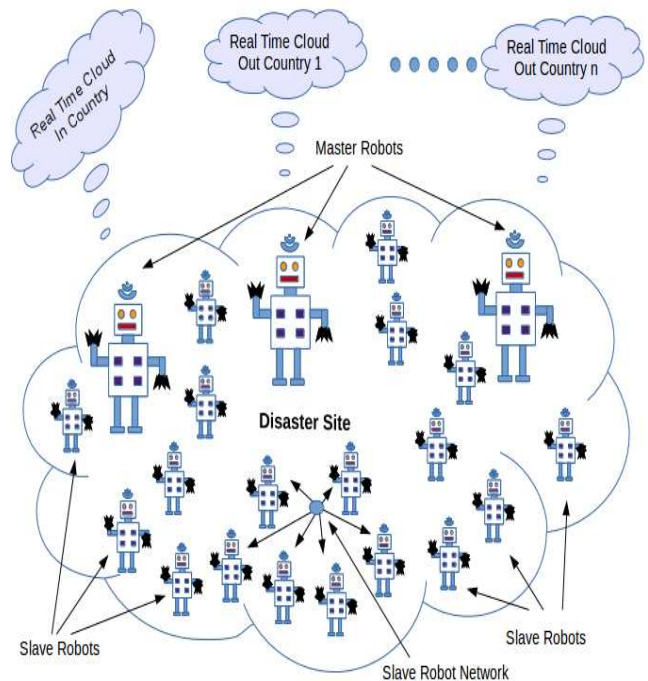


Figure 8. Disaster Control Infrastructure with Cloud Robots

REFERENCES

- [1] S. Zhang, X. Chen, S. Zhang and X. Huo, *Cloud Computing and Grid Computing 360-Degree Compared*, In IEEE Transaction On Computer Application and System Modeling (ICCAASM), Vol. 11, No. 72, October 2010.
- [2] J. J. Kuffner, *Cloud-Enabled Robots*, In IEEE-RAS International Conference on Humanoid Robots, Nashville, TN, 2010.
- [3] N. K. Jangid, *Real Time Cloud Computing*, In Proceeding National Conference on Data Management and Security, Amity University Rajasthan, India, 2011.
- [4] T. Bjerregaard and S. Mahadevan, *A Survey of Research and Practices of Network-on-Chip*, In ACM Computing Surveys, vol. 38, no. 1, March 2006.
- [5] A. K. Singh, M. Shafique, A. Kumar, and J. Hankel, *Mapping on Multi/Many-core Systems: Survey of Current and Emerging Trends*, In DAC13 Proceedings of the 50th Annual Design Automation Conference, no. 1, New York, ACM USA, 2013.

- [6] E. Carvalho, N. Calazans and F. Moraes, *Dynamic Task Mapping for MPSoCs*, In IEEE Magazine on Design and Test of Computers, vol. 27, no. 5, pp. 26-35, Copublished by the IEEE CS and the IEEE CASS, 2010.
- [7] B. Kehoe, S. Patil, P. Abbeel, K. Goldberg, *A Survey of Research on Cloud Robotics and Automation*, In IEEE Transactions on Automation Science and Engineering, vol. 12, no. 2, University of California, Berkeley, CA, USA, April 2015.
- [8] A. Silberschatz and P. B. Galvin, *Operating System Concept*, 7th ed., John Willey and Sons, 2010.
- [9] J. W. S. Liu, *Real-Time Systems*, Pearson Education, 2014.
- [10] "Real-Time Garbage Collection." [Online]. Available: <http://www.ibm.com/developerworks/java/library/j-rtj4/index.html>
- [11] "Real-Time Specification for JAVA." [Online]. Available: <http://www.rtsj.org/>
- [12] "Top 500 The List" [Online]. Available: <http://www.top500.org/lists/2015/11/>