



MULTI-CORE WITH HT TECHNOLOGY TO SOLVE DATA LAKE COMPLEX PROBLEMS

Dr. K Babu Rao¹,

¹Professor and Vice-Principal,

Department of CSE,

Vaageswari College of Engineering,

Karimnagar, Telangana, India.

Professor.kbrao@gmail.com

May 23, 2018

Abstract

Technology is taking its lead towards growth and it should be timely used properly for complex problems. Everything is available with us and it should be planned properly for better technology and its usage. This paper deals with core technology that will save time and cost. Multi-cores are not new to the society; they are taken their shape into different transitions. Now we are in dual core technology with more processors and speed. We can utilize the cores properly if we have software technology. Main objective of this paper is to utilize cores proper for specific work with interrelating them. Finally they have to be organized properly with governance of the software. HT Technology is used to select core for specific job. By selecting different cores we can gain performance and same way increase the speed.

Key Words: Multi-cores, HT Technology, speed, performance

1 Introduction

Technology is growing rapidly with lot of changes in them. Now we are in advanced technology with sophisticated resources. Computer system have coordination with the others devices and maintain them. One of critical area in computer system is the CPU or processor. There are lot improvements in the processor technology. We know single core processor, multiprocessor and now are in stage of multi-processor/multi-core as shown figure 1.

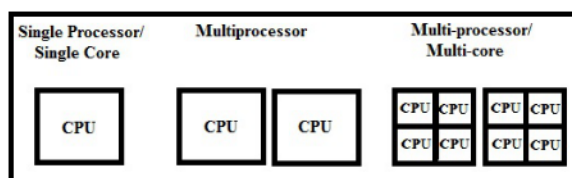


Figure 1 Three system configurations

Multi-core systems configured in an asymmetric layout where each processor has a different view of the system. One of the main challenges in programming these systems was the coordination of access to shared data by the multiple processors. The same challenge exists in today's multi-core processor systems when running under a shared memory environment. The same relative benefits of past multiprocessor systems are seen in today's multi-core. These benefits are faster execution time and increased throughput.

2 EXISTING SYSTEM

History of computers is long and it has its own milestones and it will give good and better systems to the world in coming days. Our journey started with single cores to now dual core processors. Processors play a key role in computer systems as shown in figure 1. This processor accommodate with various speeds. Now the speed range is in GHz. We have seen various stages of improvements in processor technology. Single core processor can perform less than multiple processors. Now we are in a technology where we can have processor on one chip die. This is most power achievement in processor technology. We have seen various stages of improvements

in processor technology. Single core processor can perform less than multiple processors. Now we are in a technology where we can have processor on one chip die. This is most power achievement in processor technology. Main origin for processor technology is a model proposed in a parallel processing Main origin for processor technology is a model proposed in a parallel processing technology.

3 PROPOSED SYSTEM

Now a days the chipset contains multi-cores, more than one processor in there chipset. Obviously speed can be affected by multi-cores. The problem is how we can engage them efficiently. We can make use of each processor for different usage in complex systems. It can be medical scan or surgery, airplane, weather forecasting, virtual scenarios, video games, image processing, and so on. It can be anything or field or system can be managed efficiently with multi-cores. We can have one (single), two (dual), four (quad), 6, 8, 10, 12, 16, 18, 22, 24 core processors and quad core processor is shown in below figure 2.

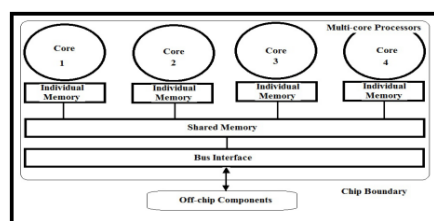


Figure 2 Multi-cores.

There are three different options to choose in using multi-core processors.

1. Do nothing
2. Multi-thread
3. Multi-task or partition

Do nothing maintain the same legacy software with no changes to accommodate multi-core processors. This option will result in

minimal performance increases because the code will not take advantage of the multiple cores and only take advantage of the incremental increases in performance offered through successive generations of improvements to the micro-architecture and the software tools that optimize for them. The third option is to multi-task or partition. Multi-tasking is the ability to run multiple processes at the same time. Partitioning is the activity of assigning cores to run specific operating systems. Multi-tasking and partitioning will reap performance benefits from multi-core processors. The second option is to multi-thread your application. Multi-threading is one of the main routes to acquiring the performance benefits of multi-core processors. Multi-threading requires designing applications in such a way that the work can be completed by independent workers functioning in the same sandbox. In multi-thread application, the workers are the individual processors cores and the sandbox represents the application data and memory. Consider the following example, the entrepreneur has some successful business and obtain funding to the hires a staff of four workers. Each of these workers is assigned different functional areas and coordinate their work so simple problems such as paying the same bill twice are avoided. Even though there are multiple workers coordinating their activities, the team is pretty able to finance the hiring of hundreds of employees. There should be proper planning to divide work effectively and requires efficient communication skills between them. Concurrency in software is a way to manage the sharing of resources used at the same time. Concurrency in software is important for several reasons like concurrency allows for the most efficient use of system resources. Efficient resource utilization is the key to maximizing performance of computing systems. An unnecessary creating dependency on different components in the system drastically lowers overall system performance. As computer architects looked at ways that processor architectures could adopt a thread-level parallelism, they realized that in many cases, the resources of modern processor were underutilized. A thread can be defined as a basic unit of CPU utilization. It contains a program counter that points to the current instruction in the stream. It contains CPU state information for the current thread. It also contains other resources such as stack. A physical processor is made up of a number of different resources including the architecture state, the general purpose CPU registers

and interrupt controller registers, caches, buses, execution units and branch prediction logic. However in order to define a thread, only the architecture state is required. A logical processor can be thus be created by duplicating this architecture space. The execution resources are then shared among the different logical processors. This technique is known as simultaneous multithreading or SMT. Intel implementation of SMT is known as Hyper Threading Technology or HT technology. HT technology makes a single processor appear from software perspective, as multiple logical processors as shown in figure 3. This allows operating systems and applications to schedule multiple threads to logical processors as they would on multiprocessor systems. Multi-core processors use chip multiprocessing (CMP). Rather than just reuse select processor resources in a single-core processor, processor manufacturers take advantage of improvements in manufacturing technology to implement two or more execution cores within a single processor. These cores are essentially two individual processors on a single die.

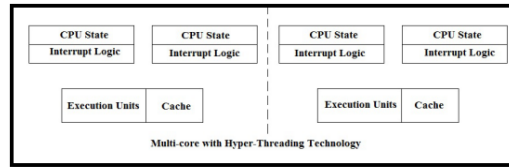


Figure 3 Multi-core with Hyper-Threading Technology.

HT Technology achieves performance gains through latency hiding. With HT Technology, in certain applications, it is possible to attain, on average, a 30 percent increase in processor throughput. In other words, in certain cases, the processor can perform 1.3 times the number of executed instructions that it could if it were running only one thread. To see a performance improvement, applications must make good use of threaded programming models and of the capabilities of Hyper-Threading Technology. Performance calculation purpose ratio is known as the speedup and characterizes how much faster a program runs when parallelized.

$$\text{Speedup}(n_t) = \frac{\text{Time}_{\text{Best_sequential_algorithm}}}{\text{Time}_{\text{Parallel_implementation}}(n_t)} (1)$$

Speedup is defined in terms of the number of physical threads (nt) used in the parallel implementation. Amdahl started with the intuitively clear statement that program speedup is a function of the fraction of a program that is accelerated and how much that fraction is accelerated.

$$Speedup = \frac{1}{(1 - Fraction_{enhanced}) + (Fraction_{enhanced}/Speedup_{enhanced})} \quad (2)$$

So, if you could speed up half the program by 15 percent, then

$$Speedup = 1/(1 - .50) + (.50/1.15) = 1/(.50 + .43) = 1.08 \quad (3)$$

This result is a speed increase of 8 percent. If half of the program is improved 15 percent then the whole program is improved by half that amount. To understand threading for your application you need to understand the following items: the design approach and structure of your application, the threading application program interface (API), the compiler or runtime environment for your application and the target platform on which your application will run. Based on HT technology we can write different programs and coordinate them for speed and fast results. We can assign different jobs or processes to threads; they will be in that assigned work till status is closed. We can assign threads to single different processors and coordinate them for merge any data.

Now a days multi-core play a key role in data related issues. Multi-core processors are used in various fields of computer related systems. Figure 4 gives detailed information of a banking model. Which contain interface software for end users, banking management system software and different databases. In figure it narrates about Data Lake, Data Lake means the data from different databases are poured in to it. Data Lake contains large datasets related to the organization. Organization can be banking system, insurance company, supermarket or any multinational company, where it contains transaction on cash, customers, employee, sales, and products. To execute large data sets we require multi-cores.

For example if we take banking system, normal banking systems used to depend upon mainframe or minicomputer or microcomputer

to execute large data sets. If the execution takes place on multi-cores then the speed and efficiency will increase. The multi-cores should be dedicated to the software. In banking system a person will have one or more saving accounts. Every person to offer credit or loan he should possess saving or current account. Before providing any credit or loan, higher officials will go through his/her saving or current account for any issues. Bank system will not offer both credit and loan at a time to a person now days. Bank offers credit card to a person, if the return payments are good then they may offer loan to a person. For knowing the details of the persons, all information related to the accounts will be send to the data lake. Because all the information are different tracks even though they are related to one person. These three tracks should be merged at one point, that point is Data Lake. The data is poured into the data lake. Based on banking software and multi-cores we can access the details of the person very fast and speed to maximum extent as shown in figure 4. Every person makes transactions every day, so there will be lot of transaction information in everybody accounts. When we have this information we can easy track the person account details whether he is genuine customer or fraud. We can have his track record very fast.

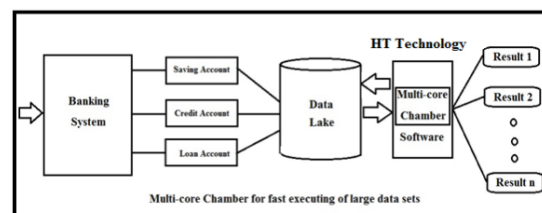


Figure 4 Multi-core Chamber for fast executing of large data sets

Based on past history we can have result 1, result 2 and result n. Result 1 details track the saving account, same way result 2 deal with saving account and credit details, result 3 deal with saving account, credit card and loan details and similarly result 'n'. We can have as many results as we can based on availability of the datasets. Searching process is a challenge in the data lake because it contains large data sets of all account holders. Here we can adopt data mining, index, trees and various searching techniques.

4 IMPLEMENTATION

Implementation process is a challenge in this context. Software should be very user friendly. Implementation can carry on with any software. Software should provide all information related to the customer, sometimes it has to detect the fraudulent. It maintains database for saving account, it maintain database for credit information and it maintain a database for loan account. These three accounts are merged to a data lake. Based on HT technology we can arrange threads to perform tasks on selected processor for execution. The bank officials can have a look on saving account, credit account and a loan account for top up of the loan. If everything is satisfactory then officials can go for top up of the loan. Algorithm: Step 1: Merge the databases to Data Lake as shown in figure 5.

$$db1 = \text{gettuples}, db2 = \text{gettuples}, \dots \quad (4)$$

$$\text{Merge} = db1 + db2 + \dots \quad (5)$$

$$\text{Datalake} = db1 + db2 + \dots \quad (6)$$

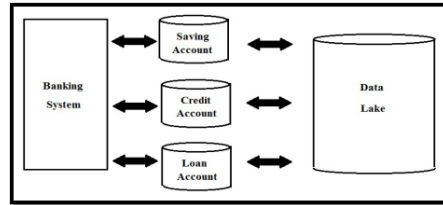


Figure 5 Merge of databases.

Step 2: All the information from the databases will be updated in the data lake. Index is used to identify the account holder with strong key. Weak entity keys will form a partial key with strong entity. Every transaction will be indexed and linked with the strong key. Likewise officials can track the transaction details. Step 3: We can have all the information about bank customer. Here we can know whether the customer paid credit card bill or not. Whether he/she deposited any amount for any payments or not vice-versa. Here we can make use of multi-cores for calculations and knowledge related to the accounts. For calculating large data sets we can use multi-cores.

5 Test results

Step 4: Based on tracking data lake we can get test results as soon as possible. We can have n number of results. Results are obtained based on the requirements. Suppose the official requires saving account details.

$$Key = \{\{tid, amount, timestamp\}, \{tid, amount, timestamp\}, \dots\} \quad (7)$$

Officials require saving account and credit account details.

$$Strongkey + Weakkey = \{\{tid, amount, timestamp, Status\}, \dots\} \quad (8)$$

Credit account details will be combination of strong and weak entity keys. It gives a set of transaction id (tid), amount to be paid, timestamp and status whether paid or not. Speed related graph is shown in below figure 6.

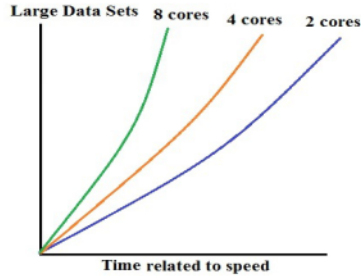


Figure 6 Speed graph on multi-cores.

6 CONCLUSION

We are well versed with core technology and advanced technology is dual core technology where we can have pairs of processors. According to cost point of view they are not too much expensive, normal people can afford it. In future we may get more cores under low cost. We are advancing in technology but poor in software. Managing software is also a complex task. Multi-core technology is

like that it can be easily used in between hardware and software. If both technologies have coordination between them, we solve so many problems. In this paper we focus on HT Technology to select cores. Threads are coordinated with the processes or jobs to solve large problems. We can use HT Technology in banking system for fast execution of the jobs or processes. We can use anywhere in the complex areas to solve complex problems. This will give good results in performance point of view and speed point of view.

References

- [1] Xuanxia Yao, Peng Geng, Xiaojiang Du. A task scheduling algorithm for multi-core processors. IEEE Xplore, 2014.
- [2] Jie Chan and William Watson III, Weizhen Mao. Multi-Threading Performance on commodity Multi-core Processors.
- [3] Pawel Gepner, Michal Filip Kowalik. Multi-core Processors: New way to Achieve High Systems. Fifth International Conference on Parallel Computing in Electrical Engineering, 2006.
- [4] Myung-Jin Baek, Jong-in Lee, Eun-Sup Sim, Hak-Jung Kim, Joo-Jin Lee. On-board management of multiple processor spacecraft system. Areospace Conference Proceedings, IEEE, 2002.
- [5] Hiroshi Inoue, Toshio Nakatani. Performance of Multi-process and multi-thread processing on multi-core SMT Processors.
- [6] Diogo Marques, Helder Duarte, Leonel Sousa, Aleksandar Ilic. Analyzing Performance of Multi-cores and Applications with cache-aware Roofline Model, IEEE Xplore, 2017.
- [7] Daniel S Bernstein, Theodore J Perkins, Shlomo Ziberstein, Lev Finkelstein. Scheduling Contract Algorithms in Multiple Processors. AAAI, 2002.
- [8] Khaled M Attia, Mostafa A El-Hosseini, Hesham A Ali. Dynamic Power Management techniques in Multi-core architectures: A Survey Study. Ain Shams Engineering Journal, September 2017.

- [9] Thomas Mezmur, Zhetao Li, Hiroo Sekiya, Nobuyoski Komuro, Young-June Choi. Efficient Thread Mapping for Heterogeneous Multicore IoT Systems. Mobile Information Systems, 2017.
- [10] Francesco Fusco, Deri Luca. High Speed Network Traffic Analysis with Commodity Multi-core Systems. Internet Measurement Conference, 2010.
- [11] Dr. Kuntam Babu Rao, Dr N Satyanaryana. Parallel Processing Shared Memory with Snoopy Cache. IJRAET, 2014.
- [12] Dr. Kuntam Babu Rao. A Novel approach for executing large datasets by a MCC Model. IEEE Xplore, 2017.