ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 5, February 2016 in association with
HEERA COLLEGE OF ENGINEERING AND TECHNOLOGY, THIRUVANANTHAPURAM
Organizes
NATIONAL CONFERENCE ON ENGINEERING FOR LIFE (NCEL – 2016)
(12th -13th February 2016)

Parallel Indexing For Unstructured Data Using OpenMP

FAIMINA B S
M.Tech in Computer Science with Specialization in Cyber Forensics and Information Security
ER&DCIIT, Trivandrum
Kerala, India riyafaimi@gmail.com

***Abstract*--** **OpenMP is a standard API for shared memory parallel programs. When we run a parallel program or a sequential program in a most modern processor which have 4 or more independent CPU cores, the instructions will only be executed in a single core and other cores will be idle and thereby waste of available resources. Here lies the importance of OpenMP, which will provide a portable standard parallel API specifically for programming shared memory multiprocessors. The use of an index is to optimize speed and performance in finding relevant documents for a search query. This paper proposes a parallel indexing methodology to index unstructured data using inverted index data structure and parallelize using OpenMP. Inverted index allows fast full text searches and is the most popular data structure used in document retrieval systems and most of the search engines.**

1. INTRODUCTION

Traditionally, software has been written for serial computation in which the problem is broken into a discrete series of instructions where instructions are executed sequentially one after another on a single processor and also only one instruction may execute at any moment in time. Parallel computing in which the simultaneous use of multiple compute resources are there to solve a computational problem by broken into discrete parts that can be solved concurrently and each part can be further broken down to a series of instructions where instructions from each part execute simultaneously on different processors and also an overall control/coordination mechanism is employed.

Parallel processing provides a cost-effective solution to this problem by increasing the number of CPUs in a computer and by adding an efficient communication system between them. The work-load can now be shared between different processors. This results in much higher computing power, performance than could be achieved with traditional single processor system. The core elements of parallel processing are CPUs. Based on a number of instruction and data streams

that can be processed simultaneously, computer systems are classified into the following four categories:

1. Single Instruction Single Data (SISD)
2. Single Instruction Multiple Data (SIMD)
3. Multiple Instruction Single Data (MISD)
4. Multiple Instruction Multiple Data (MIMD) In shared memory MIMD model, all the PEs are connected to a single global memory; all the PEs have access to this global memory also called as the tightly-coupled multiprocessor system.

A hybrid model combines more than one of the previously described programming models. Currently, a common example of a hybrid model

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 5, February 2016 in association with
HEERA COLLEGE OF ENGINEERING AND TECHNOLOGY, THIRUVANANTHAPURAM
Organizes
NATIONAL CONFERENCE ON ENGINEERING FOR LIFE (NCEL – 2016)
(12th -13th February 2016)

is the combination of the message passing model (MPI) with the threads model (OpenMP).

2. OpenMP

OpenMP (Open Multi-Processing) was first released in 1997 and is a standard Application Programming Interface (API) for writing shared memory parallel applications in C, C++ and FORTRAN. OpenMP has the advantages of being very easy to implement on currently existing serial codes and allowing incremental parallelization. It also has the advantages of being widely used, highly portable and ideally suited to multi-core architectures (which are becoming increasingly popular in every day desktop computers). OpenMP operates on a Fork and Join model of parallel execution and this is shown in Fig 5.1. All OpenMP programs begin as a single process which is called the master thread. This master thread executes sequentially until a parallel region is encountered. At this point the master thread 'forks' into a number of parallel worker threads. The instructions in the parallel region are then executed by this team of worker threads. At the end of the parallel region, the threads synchronize and join to become the single master thread again. Usually you would run one thread per processor, but it is possible to run more. Parallelization with OpenMP is specified through compiler directives which are embedded in the source code.

- OpenMP Pros:-
  - Easy to implement parallelism
  - Low latency, high bandwidth
  - Implicit Communication
  - Coarse and fine granularity
  - Dynamic load balancing
- OpenMP Cons:-
  - Only on shared memory machines
  - Scale within one node
  - Possible data placement problem
  - No specific thread order

Fig 2.1 show a sequential program is executed in an older processor having only one CPU core and in a modern processor having multiple cores. Modern processors have 4 or more independent CPU cores, but instructions will only be executed on a single core, all others will stay idle.

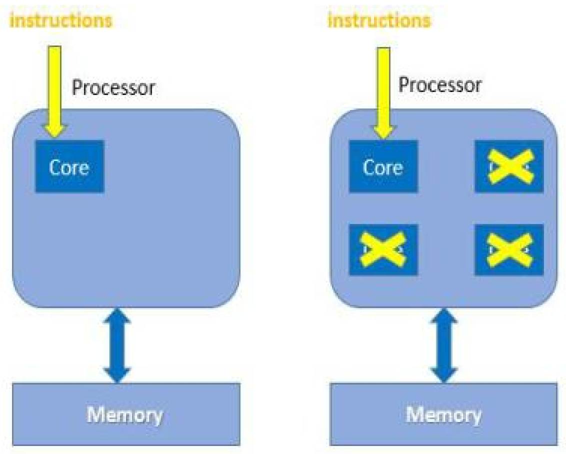

Fig 2.1 Single core and Multi core

Here lies the importance of OpenMP, in which using thread approach, the instructions are parallelized using OpenMP, and thereby available resources can be utilized efficiently, which is shown in Fig 2.2.

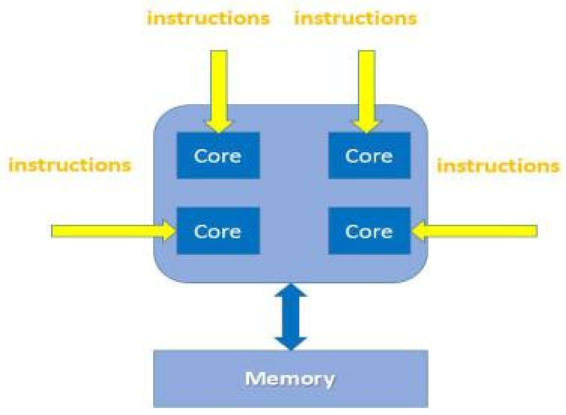

Fig 2.2 Importance of OpenMP

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 5, February 2016 in association with
HEERA COLLEGE OF ENGINEERING AND TECHNOLOGY, THIRUVANANTHAPURAM
Organizes
NATIONAL CONFERENCE ON ENGINEERING FOR LIFE (NCEL – 2016)
(12th -13th February 2016)

3. PROPOSED SYSTEM-OMP Index

OMPIndex is the name I had chosen for my system. This paper is primarily focused to design and implement an algorithm used for indexing unstructured data in a multi core computer by using the platform OpenMP. For indexing I have decided to choose inverted indexes as the data structure and parallel indexing as the methodology to follow. The data can be classified into 3 types: structured, semi-structured and unstructured data. Structured data is those data that can be defined through a relational database while semi-structured data is that through some processing we can convert the data to be defined within a relational database ie which have a row column format. More than 90% of the whole data actually belongs to unstructured data. Examples for unstructured data are text and multimedia data such as PDF, DOC, HML, Images, Audios, Videos etc. In real-world applications, users demand results quickly from a search engine-query latencies longer than a few hundred milliseconds will try a user's patience. The relevance of Indexing are in every Search engines and IR (Information Retrieval) systems. In its basic form, an inverted index consists of postings lists, one associated with each term that appears in the collection. The structure of an inverted index [4] is illustrated in Fig 6.3. A postings list is comprised of individual postings, each of which consists of a document id and a payload-information about occurrences of the term in the document.

Block Diagram:- The Indexer or here the OMP index system will produce an indexing query on a multicore computer which is shown in Fig 3.1. Using OpenMP which primarily focused on Shared Memory systems will parallelize the work to build the index onto each CPU cores and finally the index file is stored to the disk.

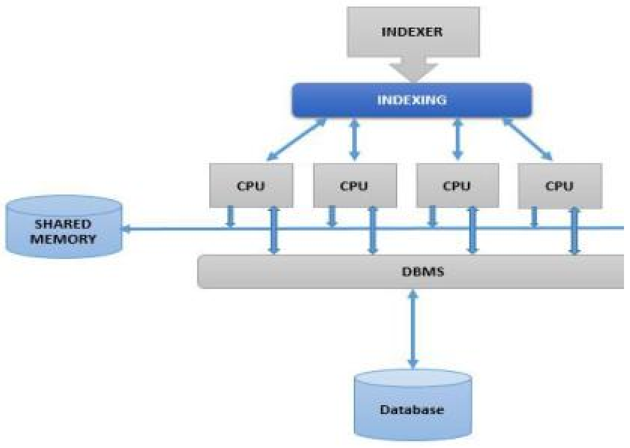

Fig 3.1 Block Diagram

4. CONCLUSION

OpenMP is really an amazing feature loaded API which can be used in many of the day to day applications, and thereby can improve their performance appreciatory. Thus I had chosen a well-used application such as Indexing Unstructured Data. Using OpenMP API we can implement the algorithm to index Unstructured Data in a shared memory architecture and thereby can compare and analyze how parallel programming through OpenMP had improved the performance of a search engine.

5. REFERENCE

1. Alessandro Camerra et al (2010) "iSAX 2.0:

   Indexing and Mining One Billion Time Series"
2. Goetz Graefe, Felix Halim (2012) "Concurrency Control for Adaptive Indexing"
3. Sanjay Kumar Sharma1 and Dr. Kusum

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 5, February 2016 in association with
HEERA COLLEGE OF ENGINEERING AND TECHNOLOGY, THIRUVANANTHAPURAM
Organizes
NATIONAL CONFERENCE ON ENGINEERING FOR LIFE (NCEL – 2016)
(12th -13th February 2016)

Gupta (2012) "Performance Analysis of Parallel Algorithms on Multi-core System using OpenMP"
4. Qiuying Bai, Chi Ma et al (2012) "A New Index Model Based on Inverted Index"
5. Alessandro Camerra, Jin Shieh et al (2013) "Beyond one billion time series: indexing and mining very large time series collections with iSAX2+"
Yang Wang, Peng Wang et al (2013) "A Data-adaptive and Dynamic Segmentation Index for Whole Matching on Time Series"
6. Naveen Garg and Dr. H.M. Rai (2013) "Bitmap Indexing Technique for Datawarehousing and Datamining"
7. Lei Yu, Ge Fu et al (2013) "TIIS: A Two-level Inverted-index Scheme for Largescale Data Processing in the Parallel Database System"
8. Kostas Zoumpatianos (2014)" Indexing for Interactive Exploration of Big Data Series"
9. Goetz Graefe · Felix Halim (2014) "Transactional support for adaptive indexing"
10. Avinash N Bhute and B.B. Meshram (2014) "Text Based Approach For Indexing And Retrieval Of Image And Video: A Review"
11. Artem Babenko and Victor Lempitsky (2015) "The Inverted Multi-Index"
12. Myeong-Seon Gil et al (2015) "Fast Index Construction for Distortion-Free Subsequence Matching in Time-Series Databases"
13. Abdullah Gani et al (2015) "A survey on indexing techniques for big data: taxonomy and performance evaluation"
14. https://computing.llnl.gov/tutorials/openMP/
15. http://www.howtogeek.com/194756/cpu-basics-multiple-cpus-cores-and-hyper-threading-explained/
16. https://computing.llnl.gov/tutorials/parallel_comp/#Models

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 5, February 2016 in association with
HEERA COLLEGE OF ENGINEERING AND TECHNOLOGY, THIRUVANANTHAPURAM
Organizes
NATIONAL CONFERENCE ON ENGINEERING FOR LIFE (NCEL – 2016)
(12th -13th February 2016)