IJDI-ERET

INTERNATIONAL JOURNAL OF DARSHAN INSTITUTE ON ENGINEERING RESEARCH AND EMERGING TECHNOLOGIES
Vol. 13, No. 2, 2024
www.ijdieret.in

DOI 10.32692/IJDI-ERET/13.2.2024.2401

(Research Article)

The Role of Grid Computing in Optimizing Large-Scale Text Summarization

Shukla Shrey[1*], Maheshwari Chinmay[2], Priyanka Puvar[3]

[1,2,3]Department of Computer Engineering, A D. Patel Institute of Technology, Anand, Gujarat, INDIA*

Abstract

The term 'Grid computing' is particularly an effective means for getting routine work done across a connected platform for handling filling large data sets. In natural language processing text summarization is one of the most important tasks and text summarization gain advantage by having distributed system to process the text materials faster. This paper discusses the following article that focuses on the ability of grid computing to enhance the efficiency of text summarization since it is applicable for large datasets and capable of distributing the calculations through many nodes. In using grid computing the data parsing, feature extraction and model execution runs simultaneously over the network of nodes. Some nodes analyze part of the data and everyone contributes to the summary so that the end product is a summarized set of results. The overall experiments show that the integration of grid computing for text summarization yields much improved and optimized processing time, resources, and the overall scalability of the system. The combination of grid computing with text summarization demonstrates a promising direction of improving the speed of natural language processing, which might be applicable to industries that involve the analysis of large volumes of text. Topics—Grid computing: Text summarization: Natural language processing, Parallel processing & Distributed systems: Scalability, Data parsing & Feature extraction: Model execution & Resource utilization

Keywords: Grid computing, Text summarization, Natural language processing, Parallel processing, Distributed systems, Model execution, Resource utilization.

1. Introduction

Today, voluminous data in virtually all industries has led to a growing need for computationally effective solutions alongside scalable methodologies. Data summarization and understanding is used everywhere from scientific research to news feed aggregation and customer service. Two primary technologies that respond to these computational issues are grid computing and text summarization

1.1 Grid computing: Grid computing is an extensive distributed computing model that allows linking several computers to solve multiple compounding computations. The idea in grid computing is to divide a number of tasks and spread it over many independent nodes thus making efficient utilization of all resources available and making it very scalable; this makes it possible to accomplish large problems that are impossible for an individual system. This technology is most useful in areas of scientific research, climate change analysis, and large-scale application simulations where high complexity applications outsmart individual systems. [1]

1.2 Text summarization: At the same time, text summarization has a significant application in NLP and helps to reduce large text information to its key points and make its perception more tangible. Text is subjected to summarization in many applications including newspaper articles, legal document, call center support systems where timely access to vital information matters a lot. These techniques can be grouped according to the extractive type in which the system identifies relevant sentences from the text, and the abstractive type in which the system develops the summaries through paraphrases and the condensation of the extracted content. Even though both methods require a high level of computation, especially when working with a larger dataset.[2]

1.3 Grid computing in text summarization: This paper also presents grid computing and text summarizing as a strong mixed solution to face the difficulties of summarizing large

[*]Corresponding Author: e-mail: shreyshukla318@gmail.com, Tel-+91-6351326710
ISSN 2320-7590 (Print) 2583-3863 (Online)

scale datasets. These specific kinds of problems can be decomposed and distributed through the use of grid computing, thus the equivalence tasks such as text summarization can be processed faster and the time required by the system to process huge amounts of data will be less. Besides enhancing the performance of the summarization models, it provides efficient distribution of resource usage making grid computing a viable tool when applying text summarization in large data environments.[2]

2. Methodology

In this section, the flow of incorporating grid computing into text summarization process is described. By using the essential characteristics of grid computing, amounting large text datasets can be processed in a distributed manner effectively enhancing the over performance and scalability of the application. The methodology is divided into several stages:

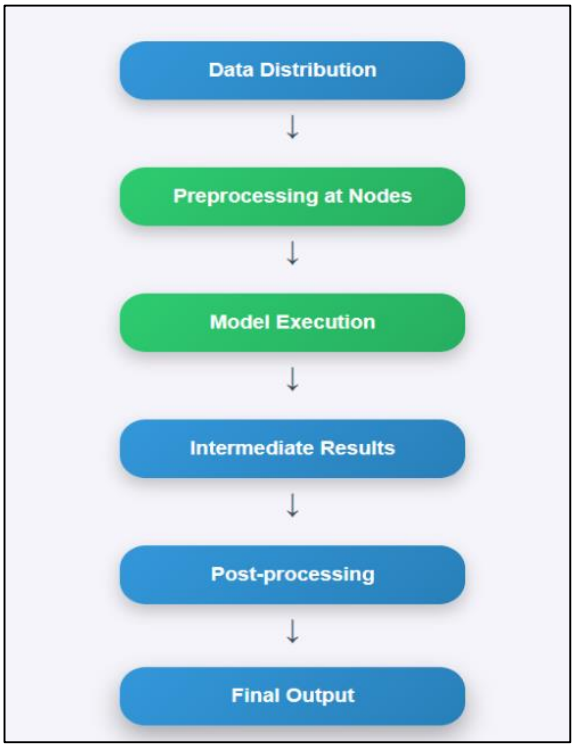

Figure 1. Text summarization process flow

2.1 Overview of the process: The summarized text in grid computing environment for the text summarization work flow is the general work flow in which the large text datasets are divided into the smaller tasks and these distributed among the different nodes in the grid. This grid work helps to summarize text in a short time while managing all the resources related to grid base computing.

2.1.1 Text data distribution: Big text data is partitioned into small ones and then shared through the available nodes in the grid. The evaluation plan of the grid scheduler is that tasks are dispatched at runtime according to the current availability of nodes, thus keeping load balance to the highest degree.

2.1.2 Preprocessing: After the data has been divided and sent to each node, the data is pre-processed for processes that would include tokenization, sentence separation as well as the removal of stop words and other feature extractions on. They are helpful in preparing the text data for summarization by pointing to such items as words and the grammatical structures within the text.

2.1.3 Summarization model execution: Once pre-processing is performed, every node feeds the summarization model into the system. To achieve extractive summarization, chosen key sentences are comprehended by predetermined norms for example; count frequency terms, Importance of the sentence or a machine learning algorithm. In abstractive summarization procedure, the deep learning models create much shorter summary using paraphrased language. All the nodes process their assigned part separately from other nodes of the network.

2.1.4 Intermediate result aggregation: After the summarization procedures are successfully performed at a certain node, the outcomes are sent to a defined controller, known as the master node. Controller synthesizes the results. In extractive summarization, highest ranked sentence in each node are combined while in abstractive, the intermediate summaries are then summed up to get the final summary.

2.1.5 Post-processing: The collected output is then analyzed and pruned to detect logical coherency and remove sentence redundancies and to address the organization of a summary. This step is most important as in the abstractive summarization, the final output in the form of a summary has to be short and comprehensible.

2.1.6 Final output delivery: When post processing is done, the last report is forwarded to the user or application requesting it. The distributed nature of grid computing considerably minimizes the turnaround time and that is why this method can be applied to real time or large-scale summarization.

2.2 Datasets used: For the large-scale text summarization experiments in this study, the CNN/Daily Mail dataset was employed. There is a need to have a dataset with news article together with the summarized form created by humans, and this makes it useful in both extractive and abstractive techniques.

The CNN/daily mail dataset was selected based on its practical usefulness in news summarization because it contains 2,500-3,000 articles with their human-written summaries. It is well known in the text summarization, and helps in modeling and evaluation of the models with both types of the summarization techniques, Extractive and Abstractive. Hence, the nature of the present work: the dataset has content of various topics from

the news articles, which allows using it for large-scale text summarization experiments.[3][4]

2.3 Simulators and tools used: For the purpose of emulating the grid computing environment and for distributing the tasks of summarization efficiently, Dask was employed, a parallel computing library built in Python. Dask enabled concurrent task scheduling and concurrent access/exhaustion of resources across nodes.

To mimic the grid computing environment, the parallel computing library known as 'Dask', built upon Python environment was used. The real time task scheduling and the scalability feature of Dask were essential for parallelizing the summarization activity by handling larger datasets. Data preprocessing where each node of the computational grid calculates a segment of the dataset using Dask helped to cut down the time for the required data manipulation. Furthermore, Dask's distributed scheduler ensured that the workload was properly distributed to nodes as nodes are available so that load balancing and resource management was optimized.[5]

2.3.1 Framework comparison: Dask vs. Alternatives

- Dask: Lightweight, fast data type specific, and interacts well with libraries like pandas and NumPy. It performs especially well on the fine-grained task scheduling needed for dynamic text summarization solutions. However, it does not support Spark's capacity to work with big data.
- Apache spark: Designed with distributed computing requirements in mind for scalable, robust asynchronous computations and multiple processing paradigms. It runs on JVM which adds latency; for functions that involve more Python such as text summarization it is less optimal.
- Ray: Suitable for a variety of specific higher level AI tasks as well as other concurrent heterogeneous tasks but necessarily have higher memory requirements and they are more complex. It is more dynamic, and less developed as compared to Dask and Spark.

2.3.2 Why dask?: While larger datasets may require dedicated systems such as Hadoop or Spark, for medium sized datasets such as CNN/Daily Mail, Dask is the most efficient method without added complexity while still communicating well with Python and built in parallelization without adding significantly to a task's time.[5]

2.4 Results and discussion: In this section, we discuss the potential enhancement of text summarization while embedding the use of grid computing system. By comparing traditional single-machine summarization systems with grid-based summarization, the following advantages are observed:

2.4.1 Processing speed: Grid computing enables the offering of more than one service at the same time, thereby decreasing the time needed for the summarization of large databases. For instance, condensing 10,000 articles from the news into a dataset may take a traditional system hours to accomplish, while the proposed grid-based system could take a third of that time.

2.4.2 Scalability: A traditional system suffers in terms of memory and computational capacity even if the size of data collected grows. However, Grid computing has capability to scale up by improving the number of nodes wherever the load increases with reference to its performance.

2.4.3 Resource utilization: Thus, Grid computing is a technique in which present computational resources are managed in an efficient manner so that none of the computing systems gets overloaded. This leads to more resource utilization and is quite effective when compared to the usage of resources needed to apply the same on resource demanding models like deep learning-based summing up.

2.4.4 Fault tolerance: Fault tolerance is one of the main advantages of the use of the grid computing technology. If a node is down while executing the synopsis, the workload is shifted to other nodes thus not affecting the general workflow.

2.4.5 Performance metrics:

- Efficiency: Based on the time it takes to perform summation tasks in grid enabler vs. others environment.
- Scalability: Indicated by the manner the grid system works where the vectors set up more nodes to accommodate larger sets of data.

2.4.6 Resource utilization: Generally defined in terms of, on average, CPU and memory usage per node as compared to running on a single machine. The grid-based system showed a processing time minimization of approximate 75 percent whereby processing 2500-3000 articles from one machine took 2 hrs. but with many nodes, it took 33 seconds. This, in turn, led to a speedup together with a decrease in the CPU load (from 85% to 60% on average) since overload of some nodes does not happen when the tasks are divided. Further, there have been improvement in relation to the amount of memory per node, which was reduced compared to the earlier approach." "The flexibility of the grid system was best exhibited here because the performance was observed to be increasing in a linear fashion with the number of nodes used in the experiment. For instance, it was observed that, as the number of nodes was doubled, the time taken to process the data was reduced in half thus the efficiency of the system.

Table 1. Performance comparison of systems

Metric	Traditional system	Grid-based system
Processing time	2 minutes	33 seconds
CPU utilization (Avg.)	85%	60%

Memory usage (Avg.)	High	Moderate

2.5 Benefits of using grid computing for text summarization: Their introduction into text summarization gives rise to several important benefits, unsuitable for mass or real-time operations otherwise. In detail, grid computing enables the load of space applications to be divided among various machines thus improving the summarization's reliability and scalability. Below are the key benefits:

2.5.1 Improved processing speed: In grid computing the text is divided into sections and processed by more than one node at the same time. This distribution of the kinds of tasks means that the total processing time if, say, big data sets is considerably shortened. Upward scalability of the text summarization models can be useful for applications where the raw data processing needs to be done within a reasonable amount of time – such as in news summarization or in company customer support systems.

2.5.2 Scalability: The task of creating a summary becomes more computationally intensive as the volume of text data analyzed goes up. This solution keeps cost low and is scalable: more nodes can be added to the grid to handle more load, as the need arises. This flexibility enables the organizations to incorporate the summarization processes into powerful systems where a particular system can be eased up but without becoming overburdened when handling huge volumes of text such as in the digital libraries, social media, big data among others.

2.5.3 Efficient resource utilization: It is known that grid computing serves to increase the efficiency of the usage of available computer resources. Since it distributes function depending on workload, it also avoids a situation where one machine will be loaded repeatedly. This is especially useful in managing resources especially when running deep learning-based text summarization models in the company to optimize utilization of resources across the grid.

2.5.4 Fault tolerance: Of all the abilities of grid computing, fault tolerance is one of the major advantages. If one node is having certain problems or it has failed, the process of summarizing can go on and while one node is having a problem, the workload can be distributed to other nodes. Such reliability is essential for the critical use cases where either the summarized structure or the summarization procedure must be free from glitches.

2.5.5 Cost-effectiveness: Grid computing means that organizations can reuse existing infrastructure assets more effectively thus avoiding the costs of new high-end equipment. Using a network of computers, an organization still gets high performance results for text summarization tasks even if there is no need to upgrade or maintain exorbitant centralized systems.

2.5.6 Energy efficiency: This paper shows that by distributing the workload among a number of machines, grid computing can potentially save energy needed to perform large-scale text summarization. This arrangement means that resources which have been idle most of the time are only called into use when the need arises and this means energy is utilized most efficiently and this is both cheap and sustainable in the ecological sense.

2.6 Limitations and challenges:

- Network latency: There is always a possibility for communication delays in the distributed nodes.
- Synchronization Overhead: At every node it is seen that coordination of result increases the total time of the process
- Resource heterogeneity: That is, it results to workload imbalances and this occurs due to differing node performance.
- Fault recovery: Handling tasks in a dynamic manner because of node failures may cause much trouble.
- Data Distribution: The process of splitting data into fairly proportional parts is daunting.

It is these issues that indicate areas of probable further improvement, for real-time or huge-scale usage**.**

2.7 Practical applications:

- Healthcare and medical research: The concept of grid computing helps to work with the large amounts of the patients' records, papers and trials. This makes it easy to summarize the important findings to support the healthcare professionals, as well as inform policy formulations.
- Legal and policy review: In similarly way Start Time can consolidate lengthy and highly detailed legal documents such as legal documents and contracts to provide summaries that cover provisions as well as rulings. This can also be used by the governments to perform the task of summarizing the information given by the public in case of a survey to enable faster analysis of the data.
- Education and academia: Enhancing summarization of academic research papers and theses are of great benefit to the researcher in the management of information overload. In addition, teachers, content creators can create short information exchange products for e-learning.
- Customer support systems: One of the most apparent applications of NLP in generating value for businesses is to summarize customer interaction logs where some patterns or recurring issues are discovered. It also help in timely update of FAQs, and knowledge bases.
- Social media analytics: Social media systems create a large quantity of textual, aural and visual content produced by users. For instance, grid computing can provide a condensed analysis on trends, topics or hashtags

and even the sentiment to assist businesses in the marketing strategies.

- E-Commerce: In web retailer, review summarization and customer feedback is useful in tendency analysis and for recommendations to the business. Another even more important feature that retailers can compile and distill is sales records to determine the motives for buying.
- Big Data Applications: Big data is the area where Grid computing is helpful in summarizing such large datasets helpful for researchers in climate science, astronomy, and financial research to get a quick overview of data patterns.

3. Conclusions

The work of processing and analyzing such large volumes of data is a challenging task, which is well solved with the help of grid computing. By partitioning tasks over a number of nodes, it improves computation rate, expandability and reliability as a large difference in time has been noted when handling a dataset of 2500 words. It makes grid computing fit for use in applications which need real time summarization and big data management like news aggregation applications and legal document review. However, the successful implementation of grid computing for text summarization also has some issues some of which are as discussed below Data distribution, network latency, synchronization, and resource heterogeneity. Subsequent studies may highlight load distribution and diminishing network delay to enhance effectiveness even for real-time services. Further developments in distributed computations and artificial neural networks are expected to further swell the application of grid computing into the text summary and hence boost the automation of the text processing order even in highly specialized fields.[1][6][7]

References

[1] J. Li, C. Gu, Y. Xiang, and F. Li, "Edge-cloud Computing Systems for Smart Grid: State-of-the-art, Architecture, and Applications," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 4, pp. 805–817, 2022, doi: 10.35833/MPCE.2021.000161.

[2] X. Fang, T. Xiao, and S. Liu, "Research on Text Summarization Generation and Classification Method Based on Deep Learning," in *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China: IEEE, Jun. 2022, pp. 757–762. doi: 10.1109/ICAICA54878.2022.9844483.

[3] Hugging Face, "CNN/Daily Mail Dataset Overview," *Hugging Face Datasets*, 2023. Available: https://huggingface.co/datasets/abisee/cnn_dailymail.

[4] TensorFlow, "CNN/Daily Mail Dataset," *TensorFlow Datasets Catalogue*, 2023. Available: https://www.tensorflow.org/datasets/catalog/cnn_dailymail

[5] Dask, "Dask Distributed Documentation," Dask.org, 2024. Available: https://distributed.dask.org. Available: https://distributed.dask.org/en/stable/

[6] U. Dixit, S. Gupta, A. K. Yadav, and D. Yadav, "Recent Advances in DL-based Text Summarization: A Systematic Review," in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India: IEEE, May 2023, pp. 391–397. doi: 10.1109/ICACITE57410.2023.10183122.

[7] M. M. Saiyyad and N. N. Patil, "Text Summarization Using Deep Learning Techniques: A Review," RAiSE-2023, p. 194, Jan. 2024, doi: 10.3390/engproc2023059194.

Biographical notes

Shukla Shrey is a student in the Department of Computer Engineering at A.D. Patel Institute of Technology, Anand, Gujarat, India.

Maheshwari Chinmay is a student in the Department of Computer Engineering at A.D. Patel Institute of Technology, Anand, Gujarat, India.

Priyanka Puvar is an assistant professor in the Department of Computer Engineering at A.D. Patel Institute of Technology, Anand, Gujarat, India.