IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 8 (August. 2018), ||V (V) || PP 55-61

A Survey on Realizing Memory-Optimized Distributed Graph Processing

Dr.R Priya [1], Sona Mary Louis [2]
Head of the Department/Associate Professor, Department of Computer Science, Sree Narayana Guru College, Coimbatore, Tamil Nadu, India[1]
M.phil Scholar, Department of Computer Science, Sree Narayana Guru College, Coimbatore, Tamil Nadu, India[2]
Coorresponding Author: Dr.R Priya [1],

Abstract: The current trend applications involve the graph data, which is increased every time. This applications includes a number of distributed graph processing systems are, (a) pregel, (b) apache giraph, (c) graphX, but this systems not include available excessive memory that is cannot provides extra memory in real-world graphs, so the task of graph processing system for distributed environments is include one demand, that demand is needed excessive memory for distributed data. This graph processing system holds unused memory representations of adjacency lists. The memory usage pattern directly expresses how to graph the first memory area in distributed graph processing system. In this paper proposes the objectives are: (a) Three compressed techniques: Its adjacency list representations are applied in every DGPS, (b) A variable-byte encoded: Its representation only out-edge weights it is used for space-efficient support of weighted graphs, (c) A tree-based compact out-edge: Its representation allowed for efficient mutations on graph elements, that is edges and vertices. This concepts availability provides number of edges and find out the space-efficiency and execution time. It to reduce the respective memory requirements for graph elements it continuously compared with the state-of-the-art methods, in this time memory-optimized methods hold the efficiency of the uncompressed structures and then it to enables the execution of algorithms for large-scale graphs include alternative structures on memory oriented errors.

Keywords: Distributed graph processing, graph compression, pregel, apache giraph, Memory optimization, graphX.

Date of Submission: 16-08-2018 Date of acceptance: 03-09-2018

I. INTRODUCTION

In the current stage all type of web applications are moreover increased in World Wide Web (WWW) space, that includes the number of web applications, social networks and some more free source that is open-source software and this oriented details such as product key, size of the executable file and some other needed information available in the web. This type of web applications are daily used one so many user to access this applications and get more benefit today. To access these applications perform some actions like upload, download, modify, edited and so on, firstly, to this systems that routinely to handle huge or volume of data that has modeled as called, “Graphs”. So, this reason by increasing graph-vertices to realization of a number of DGPS approaches. This approach is to handle the large-scale graphs using some hardware, and parallelize the execution of algorithms by dividing graphs into number of partitions. Then, it assigning vertices to machines (workers) and it follows the programming paradigm of “think like a vertex” and it is introduced with the approach “Pregel”. The term memory-optimization is allows the mining of the graph’s elements without decompression that is additional memory includes with unencoded representation. Apache giraph is used for graph search service, so improved performance and scalability. The pregel system includes the locality of reference, so to find out-edges (Weighted graphs).

II. OBJECTIVES:

In this concept mostly used for to find the edge that is, out-edges. Out-edges represents the space-efficient of vertex and the graph weights. It allows the fast mining of graph elements like edges and vertices without needs of decompression. It enables the execution of graph algorithms in memory-constrained settings. Finally, the memory management this task is completely fast that means faster execution.

Key Terms

Graph

A graph is combination of edges and vertices. Graph is covered the area that is this described term of as use the graph element and make the path between source to destination, it typically describes vertices as nodes, edges as paths, so the term are called graph. Graph is to implements the undirected and directed graph that meaning based on the concept form the mathematics, term like graph theory.

The elements of graph are edges and vertices are denoted by E and V that defines Graph G. Graph is to find smallest node use the above formula.

Graph G= (V, E)

Adjacency List

It is a collection of unordered lists and it is used to represent a finite graph that is predefined graph. Each of adjacency lists describes the set of neighbors of a vertex in the graph.

Apache Giraph

It only graphs processing on big data. In these terms applying giraph utilizes the apache hadoop's MapReduce implementation to process graphs. This concept is applied in the social media like facebook, and to analyze the performance improvements. This concept is used in to the Google graph processing system called "Pregel" and it is compared to another big graph processing libraries namely, such as "cassovary".

Graph Processing System (GPS)

GPS is an open-source system, so it is used for the large graphs, so to avoid the problems. The facility of open-source is based on scalability, error-curing and easy to program execution. It implements and included the vertex-centric computations with global computations. It optimizes the network problems. It can be implemented using a higher-level domain specific language, and executed. It is a called "Distributed system".

Memory Optimization

The memory optimization is a technique that minimizes the memory transfer that is short out the mapping. Use the cache obliviousness that means use the cache buffer it stores the recent files or address so easily mapped or routed the memory area. This concept not includes the pointer concept, so better space utilization, to get the lower levels of memory and more elements in cache line to map fast so easily finish their works.

III. LITERATURE REVIEW:

This paper [1] author has presented the concept of "GPS: A Graph Processing System", which is fully robust open-source system that is free source system, so access everyone and everyone download easily. It was developed for scalable that is change the size, fault-tolerant that is adjust the error for mapping or graphing, and easy-to-program execution of algorithms and that's oriented important steps on very large graph. This system describes and new techniques are presented for the best concept of Distributed Graph Processing System (DGPS) that is graph partitioning technique. The GPS system is similar to the Google's rights or proprietary rights. This system use GPS system like pregel system. This system includes three features are: (i). Use the API that is application programming interface, it is an extended feature that to make global computations that globalization or generalization the concept, so more easily expressed and more efficient; (ii). Use the static partitioning and a dynamic repartitioning scheme that re-assigned or re-valued vertices to various different workers during the computation. This scheme is based on the pattern of message. (iii). Adjacency lists is distributed from an optimization concept, this lists includes high-degree vertices fully collects and all nodes computed then, to improve the performance.

This system is to introduce the MapReduce Framework and hadoop that open-source implementation. This system provides the automatically measuring the extreme that is large volume of data, automatic error adjacency, and easy and simple programming interface based around implementing a set of or a group of functions.

This paper [2] author has presented the graph techniques for distributed memory graphs. The paper, "Distributed-Memory Breadth-First Search on Massive Graphs", solve the problem of traversing large graphs. The problem solving techniques that is a traversal systematic method analyzing all vertices that is, nodes and edges that is, path or bridge in a graph, in many different type of manner. The BFS (Breadth-First Search) is very important which means it serving many graph algorithms in form of building blocks. Parallel graph algorithm is applying the BFS, and then it analyzes all vertices in a graph, which means DFS (Depth-First Search) is generally sequential order. The parallel algorithm use BFS and the optimal sequential algorithm use DFS. The BFS is denotes some algorithmic symbols, are G denoted by Graph, s denoted by source vertex. G is covered the area of every vertex that is reachable from s, that is to find the path between source to destination. Number of vertices is connected in to another number of vertices through number of edges, in this form collectively called Graph (Undirected and directed graph). The BFS include one worst-case that means to analyze all edges are connected to s belongs to every vertex in the connected component. In this system aim is, (a) low computational intensity, and (b) Data access patterns. The parallel BFS is to improve and optimized in two major techniques namely, (i). Direction-optimization and (ii). Two-dimension decomposition. The first technique is described by beamer et al, then this technique reduces the number of edges by using Bottom-up algorithm into traversal, the second technique is 2D that is use the sparse adjacency matrix of graph. The 2D approach follows the formula in $p_{r\times}p_{c}$ rectangular processor grids.

This paper [3] author has presented DGPS for computation-centric. Generally, the DGPS is focus only scalability that is optimizes the inter-node communication, and load balance. It delivers the unsatisfactory graph into shared memory graph computation framework. In this paper presents the concept of “Gemini”, that is the DGPS is applies and optimize the targeting computation performance, so to build the scalability on top of efficiency. This system based on, (a). A sparse-dense signal-slot abstraction, that is hide the slotted information and to extend the hybrid push-pull computation model from shared-memory to distributed scenarios, (b). A chunk-based partitioning scheme that is small partitioning enables low-overhead scaling out designs and locality-preserving vertex accesses, (c). A dual representation of the scheme is to compress accesses to vertex indices, (d). NUMA-ware sub-partitioning for efficient intra-node memory accesses, (e). Locality-aware pieces of information that is called “chunking” and then fine-grained that is correctly analyzed data work-stealing for improving both inter-node and intra-node load balance.

This paper [4] author has presented the GPS concept using ReRAM. The previous GPS is includes poor locality and high memory bandwidth requirement. This GPA (Graph Processing Accelerators based on memory access optimizations or placing the computation logics close to memory, this system previous technique is used for significant data movements and energy consumption it motivates the hardware GPA (Graph Processing accelerators).

In this paper introduces the GRAPHR that is ReRAM-based graph processing accelerator. The GRAPHR is follows the policy of near-data processing and analyzes the availability of performing massive parallel analog operations with low physical component that is nothing but, hardware and energy cost. This technique steps are, (i). The algorithms are iterative and generally tolerate the approximation; (ii). Both probability calculation that is PageRank and collaborative filtering and graph algorithms involving integers that BFS/SSSP, are the directed error. The BFS is Breadth First Search and the SSSP is Single-Source Shortest Path algorithm. GRAPHR is a vertex program of graph algorithm. It can be expressed in 2D form that is nothing but, sparse matrix vector multiplication (SpMV). It can be efficiently and importantly performed by ReRAM crossbar, this is really uses a large set of graph algorithms. The technique GRAPHR consists of the two components are: (a) memory ReRAM and, (b) graph engine (GE). The graph computations are performed through 2D form in GEs that is, ReRAM crossbars. The matrix is already known and then the vector/matrix based on the graph computation is not offers the new graph computations, but the ReRAM offers the one opportunity to realize the massive that is huge parallelism with unexpected energy efficiency and low physical component cost that is hardware cost. First, small subgraphs processed by GEs, the aim of performing parallel operations hiding the unwanted reason for sparsity.

In this paper [5] author has presented the DGPS for Large-scale graphs. The distributed processing is includes size and irregular structure of graph computations. In this paper, presented the new concept HIPG, that is a distributed framework. It facilitates high-level programming of parallel graph algorithms, it expressing a hierarchy of distributed computations that executed and independently and managed by the user. This system is high-level framework for distributed processing of large-scale graphs, so the solution is finds all type of scaled graphs. The HIPG programs that is the framework of this paper introducing concept and is commonly, it is short and simple; but this achieves (a) good portability, (b) memory utilization, (c) performance, The graphs are needs to be partitioned between memories of multiple machines or sources and then it processed in parallel in such a distributed environment that is nothing but distributes the data through graph. This is allowed by the efficient storage of edges with the source nodes or points or vertices. The partitioning graph is a small number of edges and spanning that is splitting.

The parallelizing graph algorithms, amount of computation per graph’s node is commonly very small and communication especially different graph chunks that is small graphs. In this paper proposed in to the implementation of HIPG that is Hierarchical Parallel Graph Algorithms that only operate on the large-scale graphs. It provides an interface so to perform structure-driven distributed graph computations that are organized into a hierarchy that is inherited, coordinated that is combine by logical objects are known as “Synchronizers”. It not supports the not limited creating divide-and-conquer graph algorithms. In this, techniques are a program is already known, so it is automatically sequentially component provided by the user, so that model is a computational model, obviously. In this techniques in this orderly to perform, first it can be used the program graph algorithms, Then this technique introduce the three graph algorithm for increasing the order of complexity are namely, (i) Reachability search, (ii) Finding single-source shortest paths, (iii) Strongly connected components decomposition.

This paper [6] presented the survey of PGPS that is Parallel Graph Processing System Frameworks that has provided the truthful and correct survey deliver by author. A graph analysis is particularly developed in previous system use the complexity techniques or methods, that is complex network analysis, information retrieval and data mining and so on. In this paper introduced and describes the various many programming paradigms, models, and frameworks for GPS. But the problem is many data collections have spreads in size and leave the huge performance. In this GPS is to find the best balance between simple, user-friendly and productivity-enhancing front-ends and high-performance back-ends for analysis.

In this paper [7] author has presented by the Big-Memory Machine concept of "Ringo: Interactive Graph Analytics", which means a system for analysis of large graphs. Typically, the graphs offer a way to represented by the analyze systems of the interacting objects that is like people, proteins, and WebPages with edges between the objects denoting interactions that means friendships, physical interactions, links, etc. The term mining graph offers a valuable insight about the individual that is separate objects same as relationship. This system allowed and it is to build an easy-to-use interactive and attractive high-performance graph analytics system, so the graphs also need to the built input data, which in the form of relational tables. This system provides the rich and high functionality for manipulating that changing everyone, the raw input data tables into various different types of graphs. This system provides the many number of functions that can be applied to the constructs the graph.

It shows a single big-memory machine provides a very attractive and interactive platform for performing analytics on all. But the largest graph, that it provides excellent performance and use of compared to alternative approaches. This concept totally describes how to integrates the graph analytics with an its interactive process of trial-and-error that is bugs, the data exploration and rapid testing the techniques oriented logics, and the common data mining data processing that is workloads, that means use more data mining techniques so that data are flows.

This paper [8] author has presented the new concept based on chronological order that is related to Graph matching pattern theme. This paper, "PGX.D/Async: A Scalable Distributed Graph Pattern Matching Engine", which means the distributed querying pattern. This system introduces the querying system named as "Graph Querying". In this term Graph Querying (GQ) and pattern matching that graph mapping, it is becoming an efficient feature of graph processing. It allows us to data analysts so it to easily collects and correctly understand the information about the graph in a way or a path is similar to the best query passing system SQL for (Structured Query Language) databases. In the pattern matching is the process increasingly big and large graphs but not fit or fix in the single machine only. This graph is able to handle very large datasets, and that it is an implementation technique for pattern matching operations with asynchronous depth-first traversal that is related to DFS (Depth-First Search). This term allows a high degree of parallelism and control the memory consumption that is memory adjustments. In this querying system deals with some querying schemes like, PGQL, an SQL-like query language for property graphs, that all. In this system provides an (a) intuitive, (b) distributed, (c) in-memory pattern matching engine for very large graphs, so this pattern matching is applied in many terms to get the better way to source to destination matching pattern in the current trend.

This paper [9] author has presented the data mining based technique namely Memory cloud that the paper fully covered the "trinity: a distributed graph engine on a memory cloud". The computation of this technique performed by graph algorithms such as data driven, and requests a high degree of data access that is random data access, but it not provide the level of efficient random access required by this graph computation. Then, the memory-based approaches are usually not scale that is sizable by the reason is capacity limit of single machines. In this paper is to introduce the concept of trinity. Its general purpose is graph engine on a distributed memory cloud, that is commonly distributed data are stored in this cloud area, known. This term is optimized memory management and network oriented communication, so this concept supports the graph exploration same as efficient parallel computing. This graph access patterns are in both stages, the computation so to optimize the memory and communication and it is to provide the best performance. It supports online query processing and then offline analytics that this is both stages on large graphs with machines. The term trinity provides a high level specification language called TSL (Trinity Specification Language). Use TSL for the users to declare data schema that is overall design and the communication protocols nothing but communication oriented set of rules and regulations, which brings a great ease-of-use for the general purpose of the graph management and computing and then it shows the trinity's performance in both (Online/offline) low latency graph queries same as high throughput graph analytics on Web-scale, billion-node graphs, that concept are applied in the large number of nodes so we are get the better performance.

This paper [10] author has presented the parallel graph processing it applies in memory accelerator. This concept is, "A scalable processing-in-memory accelerator for PGPS", like Parallel Graph Processing

System. In the digital world expects the fast digital data access and need the fast data analysis. In memory big-data processing in every computer system that is increased one. In the large-scale graph processing is gaining the proper results and its applicability from social science to machine learning that is the data mining techniques are available to use in this part. The problem of this stage consider the term is, Scalable hardware (that is physical components) design that can efficiently process the large graphs in main memory that's all. In the cost-effective that is expense cost and then scalable graph processing systems. This system is to realize by build a system that's the performance increases and percentage with size of graphs that stored in the system, which is extremely related to memory bandwidth limitations. In this paper use the concept of PIM (Processing-in-memory), and it is applied to get this system solution to achieve the objective. This concept usage based to get the merit of such a new technology and to enable the memory-capacity–proportional performance. It uses the large-scale GPS that is Tesseract. Tesseract is designed by a programmable PIM accelerator (for Large-Scale GPS). In these terms concepts are based on these following points: (a). mainly available the memory bandwidth it was fully utilized by new hardware architecture. (b) The communication between different memory partitions used an efficient method also. (c) Use the unique hardware design and design a programming interface. It includes two hardware prefetchers, and it used for memory access pattern of GPS. This part is provides the programming model.

In this paper [11] author has newly presented the different and enthusiastic and better needs to satisfy this concept, its related to Disk-based Graph Processing. This paper increase the edges that are path or route based on user needs and purpose is a generic I/O optimization for Disk-based Graph processing System. In separate PC that is, single-PC is easily maps the disk-based processing of big graphs for popularity. This system is a well-designed for partition structure and minimized the random disk accesses. In this previous model use the static partitions, this system before starts the processing. But this system is cannot changed the edge or set the zero new impact on the computation vertex values. In this paper concept provides a common optimization and that removes the I/O inefficiency through dynamic partition so use dynamic layouts. The previous system use static partition so use static layout, so that do not changed one. This system use the dynamic that means changed one, so the edges are changed or increased based on needed edges. This concept satisfies the edge requirements for fast and easily access the source to destination pathways, obviously. The concept based implementation provides the optimization in GraphChi that represents out-of-core vertex-centric graph system. The out-of-core graphs systems are classified into two types it based on computation styles are, (a) vertex-centric, (b) edge-centric.

This paper [12] author has presented the concept is locality exists in GPS it is based on workload characterization on an Ivy bridge server that means. GPS is increased the application domain and communication-bound. It was introduce the three high performance algorithms (LLC-Last-Level Cache, NUMA- Non-Uniform Memory Access, MLP-Memory-Level Parallelism) for dual-socket server. It provides the simultaneous low compute and bandwidth utilization and to improve performance without requiring a new memory system, but the previous system fully utilize the memory bandwidth and increasing memory bandwidth utilization so due to decreasing the communication.

The paper [13] namely, "Distributed graph Storage and Querying System" has presented author to deliver the wonderful message for memory storage mapping through querying system. The graph database provides an efficient way to store the data and access inter-connected data. The query system is no longer for graph analysis, that situation based created the path and accessing. So, the concept is query large graphs that no longer fix in memory. If necessary to make multiple trips or ways to the storage device to filter and gather the data based on query. The I/O accesses the operations and slows down the query response time and previously fully used the graph so the specific benefits that graph databases offer. The database system show graphs that viewed as indivisible structures, but not allowed the hierarchical type of graph, because it affects the query performance or query response time for large graphs, so the higher level not accepted, but actually access the entire information from disk. Use the distributing storage is to extract the better performance, this concept is entire solution cannot provide, in the time of problem stage, this automatically cure the simple problem, because more number of simple problem makes the large problem already known so this type based solved. In this paper main purpose is to optimize the distributed graph storage system for scalable that is size changeable, and faster querying of big graph data that is large graph.

The concept of "GraphH: A processing-in-memory architecture for large-scale graph processing" has presented by author and deliver the paper [14] concept in the clear understandability reports. The graph processing system specifically to take large-scale graph processing requires the high-bandwidth of data access. So the graph computing include continuous scale so to achieve a high bandwidth on generic computing architectures, it includes the points are, local bandwidth wasted by using random access pattern, global data access is poor locality provides, choose the same vertex so heavy workload, more heavy conflicts so across processing units these are. The solution is provided by this concept based on this following steps, (i) How to design the hardware specifications based on fully utilize bandwidth of PIM devices and ensure locality that is, interconnection scheme, (ii) How to allocate the data based on schedule processing flow so to avoid the heavy conflicts and unbalanced workloads.

In this paper proposes GraphH, then the PIM architecture for GPS on the Hybrid memory cube array, to solve the problem. This architecture includes the SRAM-based On-Chip vertex buffers that is small storage, that to eliminates the local bandwidth wastes; It was introduces the concept of reconfigurable double-mesh connection to provide high global bandwidth. This algorithm use the methods are (i) index mapping Interval-block (ii) Round interval Pair, so to avoid the heavy conflicts and unbalanced, and then continuously introduced the two optimization methods, so to reduce the synchronization and reuse On-Chip data.

Table 1.0. Comparison table

Paper Number	**Techniques**	**Advantages**	**Disadvantages**
1.	MapReduce Framework and hadoop	Open-source implementation.	Some network problem.
2	Breadth-first search	Find path between source to destination	To analyze all edges are connected to s belongs to every vertex.
3	Gemini	Optimize targeting computation performance.	Hide the information.
4	ReRAM	To reduce poor locality, high memory bandwidth.	Time Consuming.
5	HIPG	Good portability, memory utilization, performance.	Only suitable for Hierarchical type of graph.
6	PGPS	Simple, user-friendly, productivity.	Apply only shortest graph.
7	Ringo	Rich functionality and manipulating.	Supports the single big-memory machine.
8	Graph Pattern matching.	Fast memory matching.	Not apply in Single machine.
9	Trinity	It works on both Online/offline stages.	Protocol problem Affected the performance.
10	PIM	Increase memory capacity.	Expensive.
11	GraphChi	It represents out-of-core vertex-centric graph system.	Minimize random disk access.
12	LLC, NUMA, MLP	Bandwidth Utilization	Low computation
13	Querying system	Faster response.	No longer fix in memory.
14	SRAM-based On-Chip	To reduce heavy conflicts and unbalanced workloads.	Requires high-bandwidth.

The above table 1.0 depicts the working methodologies of various techniques which can be used to achieve memory-optimization for Distributed graph processing system.

IV. CONCLUSION

This paper proposes the three techniques and the techniques are implemented these are known as compressed techniques, so compressed the out-edge representations for distributed graph processing. This out-edge terms like, BVEdges, IntervalResidualEdges, and IndexedBitArrayEdges. The term Variable-ByteArrayWeights is represents a variable-byte encoded of out-edge weights. The term RedBlackTreeEdges represents tree-based representations of mutations. It mainly focuses the vertex-centric model, like pregel, apache giraph. The memory optimization is used the distributed graph processing system but it only follows the pregel paradigm. The BVEdges is based on state-of-the-art graph techniques for compression, and achieves the best compression but it provides the slow access time to graph's elements is edges and vertices. Finally the RedBlackTreeEdges is equal to the HashMapEdges but memory is sufficient, and it shows the significant improvements.

REFERENCES:

[1]. Semih Salihoglu, Jennifer Widem. "*GPS: A graph processing system*".

[2]. Aydin Buluc, Scott Beamer, Kamesh Madduri, Krste Asanovic, David Patterson. "*Distributed-Memory Breadth-First Search on Massive Graphs*".

[3]. Xiaowei Zhu, Wenguang Chen, Weimin Zheng, and Xiaosong Ma. "*Gemini: A Computation-Centric Distributed Graph Processing System*".

[4]. Linghao Song, Youwei Zhuo, Xuehai Qian, Hai Li, and Yiran Chen. "*GraphR: Accelerating Graph Processing Using ReRAM*".

[5]. Elzbieta Krepska, Thilo Kielmann, Wan Fokkink, Henri Bal. "*A High-Level Framework for Distributed Processing of Large-Scale Graphs*".

[6]. Niels Doekemeijer, Ana Lucia Varbanescu. "*A Survey of Parallel Graph Processing Frameworks*". Report Number *PDS-2014-003.*ISSN 1387-2109.

[7]. Yonathan Perez, Rok Sosic, Arijit Banerjee, Rohan Puttagunta, Martin Raison, Pararth Shah, Jure Leskovec. "*Ringo: Interactive Graph Analytics on Bog-Memory Machines*". [2015: 1105-1110].

[8]. Nicholas P.Roth, Vasileios Trigonakis, Sungpack Hong, Hassan Chafi, Anthony Potter, Moris Motik, Ian Harrocks. "*PGX.D/Async: A Scalable Distributed Graph Pattern Matching Engine*".

[9]. Bin Shao, Haixun Wang, Yatao Li. "*Trinity: A Distributed Graph Engine on a Memory Cloud*".

[10]. Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, Kiyoung Choi. "*A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing*".

[11]. Keval Vora, Guoqing Xu, Rajiv Gupta. "*Load the Edges You Need: A Generic I/O Optimization for Disk-based Graph Processing*". [June 22-24, 2016],*978-1-931971-30-0.*

[12]. Scott Beamer, Krste Asanovic, David Patterson. "*Locality Exists in Graph Processing: Workload Characterization on an Ivy Bridge Server*".

[13]. Janani Balaji. "*Distributed Graph Storage And Querying System*". [2016].

[14]. Guohao Dai, Tianhao Huang, Yuze Chi, Jishen Zhao, Guangyu Sun, Yongpan Liu, Yu Wang, Yuan Xie, Huazhong Yang. "*GraphH: A Processing-in-Memory Architecture for Large-scale Graph Processing*".

IOSR Journal of Engineering (IOSRJEN) is UGC approved Journal with Sl. No. 3240, Journal no. 48995.

Dr.R Priya A Survey On Realizing Memory-Optimized Distributed Graph Processing." IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 8, 2018, pp. 55-61.