ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Retrieving the Data Based on User Queries in Memory Management System Using Scheduling Techniques

VIJAYAKUMAR G[1], SABARINATHAN P[2]
[1]PG Scholor,Dept of CSE,Pavendhar Bharathidasan College of Engineering and Technology,Trichy,India
[2]Assistant professor,Dept of CSE,Pavendhar Bharathidasan College of Engineering and Technology,Trichy,India

ABSTRACT: In the concept of big facts running, Big data is a broad term for figures set as a result great or compound that usual data processing applications be insufficient. In big data, RDBMS are used to maintain the memory management processing efficiently. It contains two processes such as storing information and retrieving information from memory. The retrieved information is retrieved from two location such that database (distributed), or cache memory. Same like that, if the information is less than the limited size, it will store in database. Otherwise it will store in cache memory. Regarding User query processing the information is retrieved from the available servers.challenge take in study, incarcerate, hunt, division, cargo space, transport, apparition, and in arrange privacy. The term often refers simply to the use of predictive analytics or other certain advanced methods to extract value from records, and not often to a fussy amount of information set. The main idea of this projected occupation be spread storage space arrangement and Cache remembrance. During the data meting out, its check the size of the file whether if its inside the extent wealth its stock up on the folder if its exceeds means its stored on the collection, During user handing out the data its retrieves the data since the together recollection administration user can download the documents by means of near mec hanism (revise DA).as I introduce two algorithms are multistage DA algorithm and online sheduling algorithm that`s used to efficiently store the data in both distributed data base and cache memory not only manage the memory but also download purpose is easy to using these kinds of algorithm.

KEYWORDS: Primary memory, DRAM, relational databases, distributed databases, query processing

I. INTRODUCTION

In big data, RDBMS are used to maintain the memory management processing efficiently. It contains two processes such as storing information and retrieving information from memory. The retrieved information is retrieved from two location such that database (distributed), or cache memory. Same like that, if the information is less than the limited size, it will store in database. Otherwise it will store in cache memory. Regarding User query processing the information is retrieved from the available servers. Cloud computing is a model for enabling ubiquitous, convenient, and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

There are two main categories of cloud infrastructure: public cloud and private cloud. To take advantage of public clouds, data owners must upload their data to commercial cloud service providers which are usually considered to be semi trusted, that is, honest but curious. That means the cloud service providers will try to

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

find out as much secret information in the users' outsourced data as possible, but they will honestly follow the protocol in general. Traditional access control techniques are based on the assumption that the server is in the trusted domain of the data owner, and therefore an omniscient reference monitor can be used to enforce access policies against authenticated users. However, in the cloud computing paradigm this assumption usually does not hold, and therefore these solutions are not applicable. There is a need for a decentralized, scalable, and flexible way to control access to cloud data without fully relying on the cloud service providers.

Data encryption is the most effective in regard to preventing sensitive data from unauthorized access. In traditional public key encryption or identity-based encryption systems, encrypted data is targeted for decryption by a single known user. Unfortunately, this functionality lacks the expressiveness needed for more advanced data sharingBig data is a buzzword, or catch-phrase, used to describe a massive volume of both structured and unstructured data that is so large it is difficult to process using traditional database and software techniques.

The volume of data is too big or it moves too fast or it exceeds current processing capacity. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. This data is big data**.** The unacceptable performance was initially encountered by Internet companies such as Amazon, Google, Face book and Twitter, but is now also becoming an obstacle for other companies/organizations which desire to provide a meaningful real-time service (e.g., real-time bidding, advertising, social gaming).

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

II. RELATED WORK

In Michael Vrable, Stefan Savage, Geoffrey M. Voelker et al Presents Cumulus is a system for efficiently implementing file system backups over the Internet, taking advantage of the growing availability of cheap storage options available online. Cloud service offerings such as Amazon's Simple Storage Service (S3), a part of Amazon Web Services, offer cheap storage at a fixed cost per gigabyte and are appealing for backup, since they provide an easy way to safely store data off-site. There are pre-packaged online services specifically built for backup, such as Mozy and Carbonite. Cumulus explores the other end of the design space: building on top of a very generic cloud storage layer, an example of what we refer to as building on the "thin cloud." Using a generic, minimalist interface means that Cumulus is portable to virtually any online storage service the client implements all application logic.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

In D. Lomet, and S. Sengupta et al presents As the world moves to digital storage for archival purposes, there is an increasing demand for systems that can provide secure data storage in a cost-effective manner. By identifying common chunks of data both within and between files and storing them only once, deduplication can yield cost savings by increasing the utility of a given amount of storage. Unfortunately, deduplication exploits identical content, while encryption attempts to make all content appear random; the same content encrypted with two different keys results in very different cipher text
.

In John R. Douceur, Atul Adya, William J. Bolosky, Dan Simon, Marvin Theimer et al Presents The Farsite distributed file system provides availability by replicating each file onto multiple desktop computers. Since the replication consumes significant storage space, it is important to reclaim used space where possible. Measurement of over 500 desktop file systems shows that nearly half of all consumed space is occupied by duplicate files. present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication.

In Austin T.Clements Irfan Ahmad Murali Vilayannur Jinyuan Li et al presents File systems hosting virtual machines typically contain many duplicated blocks of data resulting in wasted storage space and increased storage array cache footprint. Deduplication addresses these problems by storing a single instance of each unique data block and sharing it between all original sources of that data. While deduplication is well understood for file systems with a centralized component, we investigate it in a decentralized cluster file system, specifically in the context of VM storage. Each host periodically and independently processes the summaries of its locked files, merges them with a shared index of blocks, and reclaims any duplicate blocks.

In William J. Bolosky, John R. Douceur, David Ely, And Marvin Theimer et al presents consider an architecture for a serverless distributed file system that does not assume mutual trust among the client computers.It provides security, availability, and reliability by distributing multiple encrypted replicas of each file among the client machines. To assess the feasibility of deploying this system on an existing desktop infrastructure,measure and analyze a large set of client machines in a commercial environment. In particularly,measure and report results on disk usage and content; file activity; and machine uptimes and lifetimes,and loads.conclude that the measured desktop infrastructure would passably support, providing availability on the order of one unfilled file request per user per thousand days.

III. PROPOSED ALGORITHM

ALGORITHM DESCRIPTION

REVISED DA ALGORITHM

Revised DA algorithm guaranteed to find a weakly stable matching for a given problem. Whenever a job is rejected, any less preferable jobs will not be accepted by a machine, even if it has enough capacity to do so.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

The algorithm enters a propose-reject procedure. Whenever there are free jobs that have machines to propose to, we randomly pick the jobs.

MULTISTAGE DA ALGORITHM

Multistage DA algorithm iteratively find a better weakly stable matching with respect to jobs.The blocking job is removed from its previous machine, so that it can make new offers to machines that have rejected it before. This ensures that the algorithm does not produce new type with blocking pairs. At each stage, we Revised DA is proposed with the selected set of proposing jobs and the entire set of machines with updated capacity.

ONLINE SCHEDULING ALGORITHM

In online scheduling the decisions regarding how to schedule tasks are done during the runtime of the system. The scheduling decisions are based on the tasks priorities whichare either assigned dynamically or statically. Static priority driven algorithms assign fixed priorities to the tasks before the start of the system. Dynamic priority driven algorithms assign the priorities to tasks during runtime. An online algorithm is forced to make decisions that may later turn out not to be optimal, and the study of online algorithms has focused on the quality of decision-making that is possible in this setting. Online VM placement develops systems to predict the dynamic resource demand of VMs and guide the placement process considers minimizing the long-term routing cost between VMs.

CORE TECHNOLOGIES

DATA STORAGE SYSTEM

A distributed data store is a computer network where information is stored on more than one node, often in a replicated fashion. It is usually specifically used to refer to either a distributed database where users store information on a number of nodes, or a computer network in which users store information on a number of peer network nodes.

CACHE SYSTEM

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Cache memory, also called CPU memory, is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. This memory is typically integrated directly with theCPU chip or placed on a separate chip that has a separate bus interconnect with the CPU. The basic purpose of cache memory is to store program instructions that are frequently re-referenced by software during operation.

Fast access to these instructions increases the overall speed of the software program. As the microprocessor processes data, it looks first in the cache memory; if it finds the instructions there (from a previous reading of data), it does not have to do a more time-consuming reading of data from larger memory or other data storage devices.

Most programs use very few resources once they have been opened and operated for a time, mainly because frequently re-referenced instructions tend to be cached. Cache memory is fast and expensive. Traditionally, it is categorized as "levels" that describe its closeness and accessibility to the microprocessor:

Level 1 (L1) cache is extremely fast but relatively small, and is usually embedded in the processor chip (CPU).

Level 2 (L2) cache is often more capacious than L1; it may be located on the CPU or on a separate chip or coprocessor with a high-speed alternative system bus interconnecting the cache to the CPU, so as not to be slowed by traffic on the main system bus.

Level 3 (L3) cache is typically specialized memory that works to improve the performance of L1 and L2. It can be significantly slower than L1 or L2, but is usually double the speed of RAM. In the case of multi core processors, each core may have its own dedicated L1 and L2 cache, but share a common L3 cache. When an instruction is referenced in the L3 cache, it is typically elevated to a higher tier cache.

DATA PROCESSING SYSTEM

Data store in database and the file size is large than available size of the memory then automatically store in catch memory. Files are search from original location and download from virtual memory and cache memory.In this process, the download file is capture and collect all the relevant detail about the file size, date of creation, modification, read/write provision, execution etc. The first process is file size is match with the resource provision. If the file size is larger than the resource provision capacity, then error message is display. Using the alert message display the appropriate information

IV.PSEUDO CODE

Step1: A Multistage graph G=(V,E) is a directed graph in which vertices are partitioned into k>=2 disjoint set(set Vi) where[1 <=i<=k].In addition,if(u,v) is an edge E then u€vi,v€vi+1.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Step2: Let c(I,j) be the cost of edge(I,j).The cost of a path from(S to T) is the sum of costs of the edges on the path.

Step3: Ihe multistage graph problem is to find the minimum cost path from “S” to “T”.

Step4: The value on the edges are called cost of edges.

Cost(i,j)=min{c(j,l)+cost(i+1,l)}

l€vi+1 (i.e)cost(i,j)=cost(levels,no of nodes at the level)

C is weight assigned on each edge.

K is the no of stages.

V.EXPERIMENTAL RESULTS

User registration:

First up all to register the user for giving all the details(name,username,password,d.o.b,phone number,e-mail-id).if the registration is completed to send the security key for corresponding user email id.

Fig.1.user registration

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

after next to perform the login function enter the username and password then finally enter the securiy key code from user e-mail id then complete the login function.next to perform specific function for user account.if you don't know the security code did not to be processed.

Search:

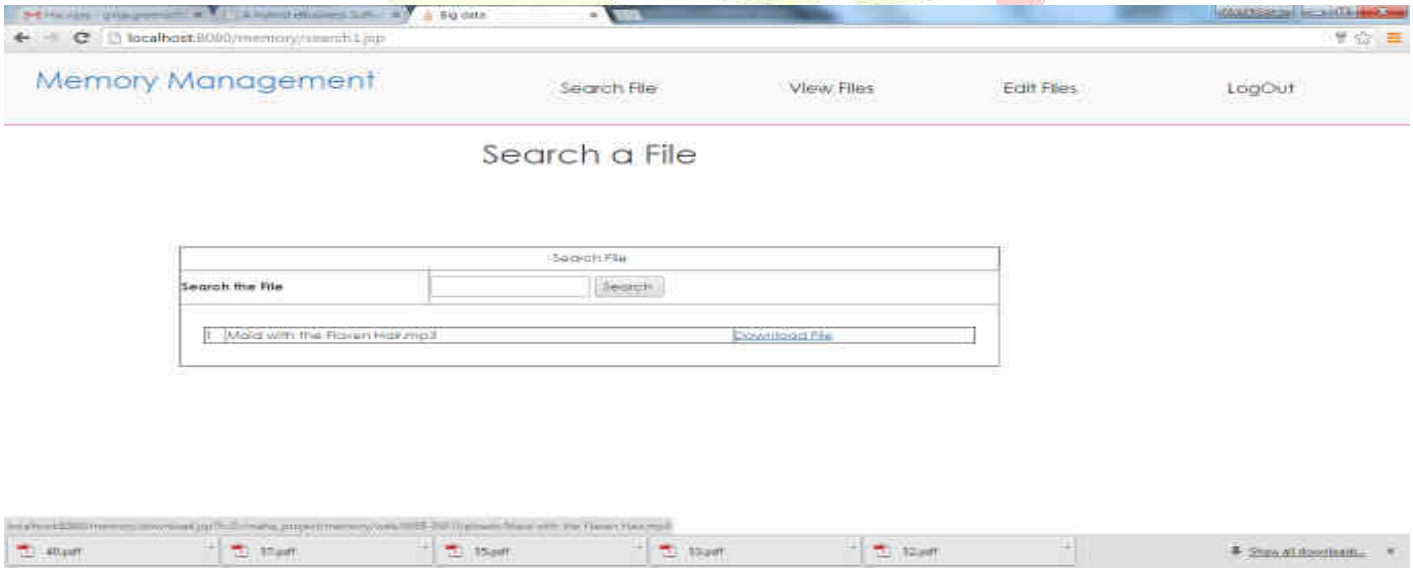

Fig.2.search a files

Resource provision:

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

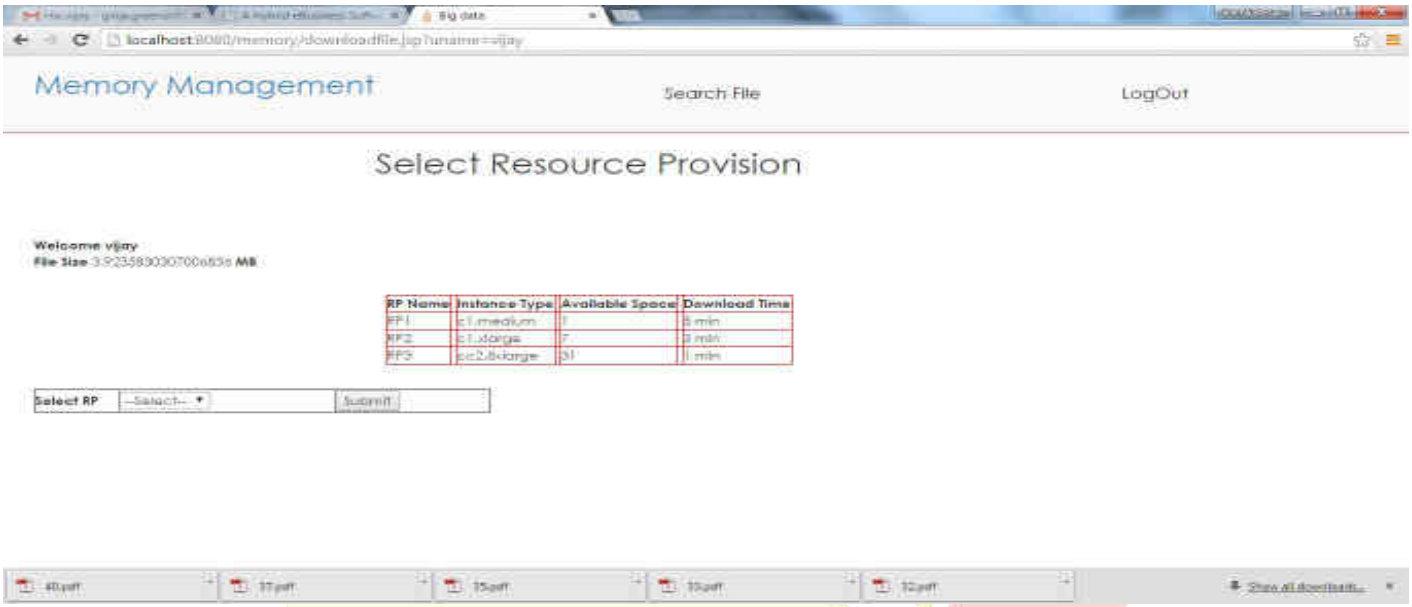

Fig.3.download the files

next one is search a file the admin upload the files,the user download the files.To search the files if the file available to display,otherwise to send the replay for not available for the data.if the file is available to download using select resource provision technique(virtual machine)for user choice,the resource provision having the capability to download the file It can be processed otherwise to send the message for to select another resource provision.

QUALITATIVE COMPARISION

The title is entitled as “QUALITIATIVE COMPARISION”. In existing one “Revised da” algorithm is used. The proposed system that is phase II two algorithm’s are used. That is multi-stage and online scheduling

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

algorithm. So compare to the existing system it is better performance and more security.

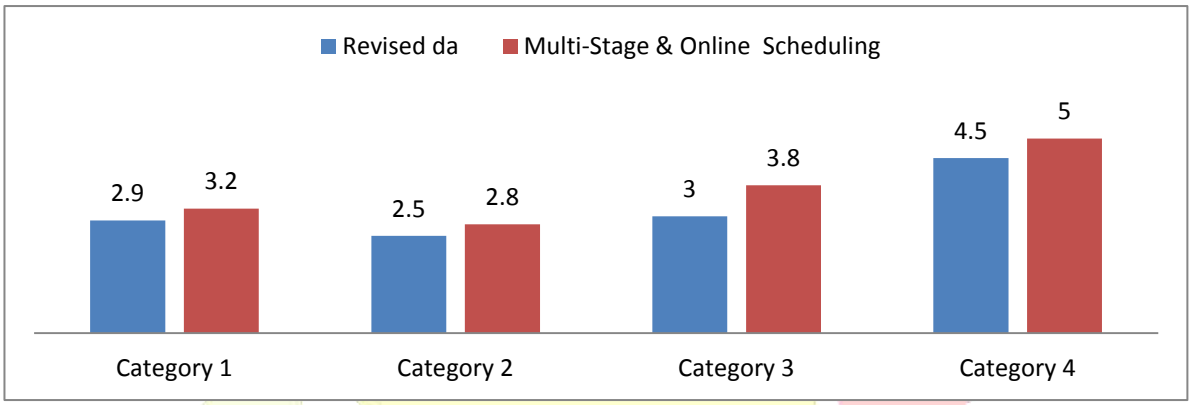

Fig.4.qualitative comparision

VI .CONCLUSION AND FUTURE WORKS

A memory becomes the new disk, processing the query in mermory management system using scheduling techniquies in big data becomes increasingly interesting for both academy and industry. Shifting the data storage layer from disks to main memory can lead to improvement in terms of response time and throughput. When data access becomes faster, every source of overhead that does not matter in traditional disk based systems, may degrade the overall performance significantly.The shifting prompts a re-thinking of the design of traditional systems, especially for databases,query processing, fault-tolerance, etc. The future direction is based on file processing using, the virtual machine. The user can download the files by using the virtual machine.The Virtual machine implement the Online Scheduling algorithm and Multistage DA Algorithm. In online scheduling the decisions regarding how to schedule tasks are done during the runtime of the system. The scheduling decisions are based on the tasks priorities which are either assigned dynamically or statically.Static priority driven algorithms assign fixed priorities to the tasks. Dynamic priority driven algorithms assign the priorities to tasks during runtime. Multistage DA algorithm iteratively finds a better weakly stable matching with respect to job.

REFERENCES

[1] S. Robbins, "RAM is the new disk," InfoQ News, Jun. 2008.
[2] J. Ousterhout, P. Agrawal, D. Erickson, C. Kozyrakis, J. Leverich, D. Mazi_eres, S. Mitra, A. Narayanan, G. Parulkar, M. Rosenblum, S. M. Rumble, E. Stratmann, and R. Stutsman, "The case for RAMClouds: Scalable high-performance storage entirely in dram," ACM SIGOPS Operating Syst. Rev., vol. 43, pp. 92–105, 2010.
[3] F. Li, B. C. Ooi, M. T. E. Ozsu, and S. Wu, "Distributed data management using MapReduce," ACM Comput. Surv., vol. 46, pp. 31:1–31:42, 2014. [4] HP. (2011).
[4]Vertica systems [Online]. Available: http://www. vertica.com
[5] Hadapt Inc.. (2011). Hadapt: Sql on hadoop [Online]. Available:
[6] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive: A warehousing solution over a map-reduce framework," in Proc. VLDB Endowment, vol. 2, pp. 1626–1629, 2009.
[7] Apache. (2008). Apache hbase [Online]. Available: http://hbase. apache.org/
[8] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford, "Spanner: Google's globallydistributed database," in Proc. USENIX Symp. Operating Syst. Des. Implementation, 2012, pp. 251–264.
[9] S. Alsubaiee, Y. Altowim, H. Altwaijry, A. Behm, V. R. Borkar, Y. Bu, M. J. Carey, I. Cetindil, M. Cheelangi, K. Faraaz, E. Gabrielova, R. Grover, Z. Heilbron, Y. Kim, C. Li, G. Li, J. M. Ok, N. Onose, P. Pirzadeh, V. J. Tsotras, R. Vernica, J. Wen, and T. Westmann, "Asterixdb: A scalable, open source BDMS," in Proc. Very Large Database, pp. 1905–1916, 2014.
[10] MySQL AB. (1995). Mysql: The world's most popular open source database [Online]. Available: http://www.mysql.com/
[11] Apache. (2008). Apache cassandra [Online]. Available: http:// cassandra.apache.org/
[12] Oracle. (2013). Oracle database 12c [Online]. Available: https:// www.oracle.com/database/index.html
[13] Neo Technology, "Neo4j - the world's leading graph database," 2007. [Online]. Available: http://www.neo4j.org/
[14] Aurelius. (2012). Titan—distributed graph database [Online]. Available: http://thinkaurelius.github.io/titan/
[15] A. Kyrola, G. Blelloch, and C. Guestrin, "Graphchi: Large-scale graph computation on just a pc," in Proc. 10th USENIX Conf. Operating Syst. Des. Implementation, 2012, pp. 31–46.