An Overview on the Structure and Applications for Business Intelligence and Data Mining in Cloud Computing

A. Fernández[1], S. del Río[2], F. Herrera[2], and J.M. Benítez[2]

[1] Dept. of Computer Science, University of Jaén, Jaén, Spain
alberto.fernandez@ujaen.es
[2] Dept. of Computer Science and Artificial Intelligence,
CITIC-UGR (Research Center on Information and Communications Technology),
University of Granada, 18071 Granada, Spain
saradelriogarcia@gmail.com, {herrera,jmbs}@decsai.ugr.es

Abstract. Cloud Computing is a new computational paradigm which has attracted a lot of interest within the business and research community. Its objective is to integrate a wide amount of heterogeneous resources in an online way to provide services under demand to different types of users, which are liberated from the details of the inner infrastructure, just concentrating on their request of resources over the net. Its main features include an elastic resource configuration and therefore a suitable framework for addressing scalability in an optimal way. From the different scenarios in which Cloud Computing could be applied, its use in Business Intelligence and Data Mining in enterprises delivers the highest expectations. The main aim is to extract knowledge of the current working of the business, and therefore to be able to anticipate certain critical operations, such as those based on sales data, fraud detection or the analysis of the clients' behavior. In this work, we give an overview of the current state of the structure of Cloud Computing for applications on Business Intelligence and Data Mining. We provide details of the layers that are needed to develop such a system in different levels of abstraction, that is, from the underlying hardware platforms to the software resources available to implement the applications. Finally, we present some examples of approaches from the field of Data Mining that had been migrated to the Cloud Computing paradigm.

1 Introduction

The Cloud Computing infrastructure has its origins in the concept of grid computing, which has the aim of reducing computational costs and to increase the flexibility and reliability of the systems. However, the difference between the two lies in the way the tasks are computed in each respective environment. In a computational grid, one large job is divided into many small portions and executed on multiple

L. Uden et al. (Eds.): 7th International Conference on KMO, AISC 172, pp. 559–570.
springerlink.com

machines, offering a similar facility for computing power. On the other hand, the computing cloud is intended to allow the user to obtain various services without investing in the underlying architecture and therefore is not so restrictive and can offer many different services, from web hosting, right down to word processing [3].

A Service Oriented Architecture (SOA) [27] is one of the basis of Cloud Computing. This type of system is designed to allow developers to overcome many distributed enterprise computing challenges including application integration, transaction management and security policies, while allowing multiple platforms and protocols and leveraging numerous access devices and legacy systems [2]. We can find some different services that a Cloud Computing infrastructure can provide over the Internet, such as a storage cloud, a data management service, or a computational cloud. All these services are given to the user without requiring them to know the location and other details of the computing infrastructure [11].

One of the successful areas of application for Cloud Computing is the one related to Business Intelligence (BI) [1, 18]. From a general perspective, this topic refers to decision support systems, which combines data gathering, data storage, and knowledge management with analysis to provide input to the decision process [24] integrating data warehousing, Data Mining (DM) and data visualization, which help organizing historical information in the hands of business analysts to generate reporting that informs executives and senior departmental managers of strategic and tactical trends and opportunities.

No need to say that the processing of a high amount of data from an enterprize in a short period of time has a great computational cost. As stated above, with these constraints a new challenge for the research community comes out with the necessity to adapt the systems to a Cloud Computing architecture [26]. The main aim is to parallelize the effort, enable fault tolerance and allowing the information management systems to answer several queries in a wide search environment, both in the level of quantity of information as for the computational time.

Along this contribution, we will first study the common structure of this type of models in order to face DM problems. Specifically, we will describe a standard infrastructure from the Cloud Computing scenario, presenting the different layers that must be taken into account to implement the management of the data, the parallel execution engine and the query language [4, 8].

Apart from the specific application of BI, the possibilities that the paradigm of Cloud Computing offers to DM processes with the aid of cloud virtualization is quite significative. This issue grows in relevance from the point of view of the parallelization of high computational cost algorithms, for example those methodologies based on evolutionary models [22, 32, 31]. Cloud Computing platforms such as MapReduce [7] and Hadoop (`http://hadoop.apache.org`) are two programming models which helps the developers to include their algorithms into a cloud environment. Both systems have been designed with two important restrictions: first, clouds have assumed that all the nodes in the cloud are co-located, i.e. within one data centre, or that there is relatively small bandwidth available between the geographically distributed clusters containing the data. Second, these clouds have assumed that

individual inputs and outputs to the cloud are relatively small, although the aggregate data managed and processed are very large.

In the last years, many standard DM algorithms have been migrated to the cloud paradigm. In this work we will present several existing proposals that can be found in the literature that adapt standard algorithms to the cloud for making them high efficient and scalable. The growing number of applications in real engineering problems, such as computer vision [34], recommendation systems [19] or Health-care systems [28] show the high significance of this approach.

This contribution is arranged as follows. In Section 2 we introduce the main concepts on Cloud Computing, including its infrastructure and main layers. Next, Section 3 presents the architecture approaches to develop BI solutions on a Cloud Computing platform. The programming models for implementing DM algorithms within this paradigm, together with some examples, is shown in Section 4. Finally, the main concluding remarks are given in Section 5.

2 Basic Concepts on Cloud Computing

We may define an SOA [27] as an integration platform based on the combination of a logical and technological architecture oriented to support and integrate all kind of services. In general, a "Service" in the framework of Cloud Computing is a task that has been encapsulated in a way that it can be automated and supplied to the clients in a consistent and constant way.

The philosophy of Cloud Computing mainly implies a change in the way of solving the problems by using computers. The design of the applications is based upon the use and combination of services. On the contrary that occurs in more traditional approaches, i.e. grid computing, the provision of the functionality relays on this use and combination of services rather than the concept of process or algorithm.

Clearly, this brings advantages in different aspects, for example the scalability, reliability, and so on, where an application, in the presence of a peak of resources' demand, because of an increase of users or an increase of the data that those provide, can still give an answer in real time since it can get more instances of a determinate service; the same occurs in the case of a fall of the demand, for which it can liberate resources, all of these actions in a transparent way to the user.

The main features of this architecture are its loose coupling, high inter-operativity and to have some interfaces that isolate the service from the implementation and the platform. In an SOA, the services tend to be organized in a general way in layers or levels (not necessarily with strict divisions) where normally, some modules use the services that are provided by the lower levels to offer other services to the superior levels. Furthermore, those levels may have different organization structure, a different architecture, etc.

There exists different categories in which the service oriented systems can be clustered. One of the most used criteria to group these systems is the abstraction level that offers to the system user. In this manner, three different levels are often distinguished , as we can observe in Figure 1. In the remainder of this section, we

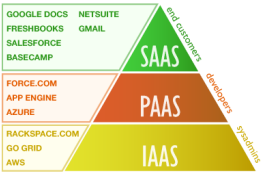

Fig. 1 Illustration of the layers for the Services Oriented Architecture

will first describe each one of these three levels, providing the features that defines each one of them and some examples of the most known systems of each type. Next we will present some technological challenges that must be taken into account for the development of a Cloud Computing system.

2.1 Infrastructure as a Service (IaaS)

IaaS is the supply of hardware as a service, that is, servers, net technology, storage or computation, as well as basic characteristics such as Operating Systems and virtualization of hardware resources [16]. Making an analogy with a monocomputer system, the IaaS will correspond to the hardware of such a computer together with the Operating System that take care of the management of the hardware resources and ease the access to them.

The IaaS client rents the computational resources instead of buying and installing its own data center. The service usually is billed based in its actual usage, so it has the advantage that the client pays for what it uses and it uses what he needs in each moment. Also, according to the dynamical scaling associated to Cloud Computing, in the case of low loads of work it uses (and pays for) less resources, and in the presence of a higher resources demand, IaaS can provide them to attend the punctual necessities of that client, being this task done in real time. It is also frequent that the contract of the service includes a maximum that the user cannot exceed.

One kind of typical IaaS clients are scientific researchers and technicians, which thanks to the IaaS and the wide volume of infrastructure that they offer as a service, they can develop tests and analysis of the data in a level that could not be possible without the access to the this big scale computational infrastructure.

2.2 Platform as a Service (PaaS)

At the PaaS level, the provider supplies more than just infrastructure, i.e. an integrated set of software with all the stuff that a developer needs to build applications, both for the developing and for the execution stages. In this manner, a PaaS provider does not provide the infrastructure directly, but making use of the services of an IaaS it presents the tools that a developer needs to, having an indirect access to the IaaS services and, consequently, to the infrastructure [16].

If we follow the analogy set out in the previous subsection relative to a monocomputer system, the PaaS will correspond to a software layer that enables to develop components for applications, as well as applications themselves. It will be an Integrated Developer Environment, or a set of independent tools, that allows to develop an engineering software problem in all its stages, from the analysis and model of the problem, the design of the solution, its implementation and the necessary tests before carrying out the stage of deployment and exploitation. In the same manner, a programming language that counts with compilers and libraries for the different Operating Systems will allow that the same application can be deployed in different systems without the necessity of rewrite any piece of code.

2.3 Software as a Service (SaaS)

In the last level we may find the SaaS, i.e. to offer software as a service. This was one of the first implementations of the Cloud services, along with the gaining in importance of the Internet usage. It has its origins in the host operations carried out by the Application Service Providers, from which some enterprises offered to others the applications known as Customer Relationship Managements [9].

Throughout the time, this offer has evolved to a wide range of possibilities, both for enterprises and for particular users. Regarding the net support, although these services are performed through the Internet, as it provides the geographical mobility and the flexibility needed, a simple exchange of data in this manner will not assure the privacy of them. For this reason, Virtual Private Networks are often employed for this aim, as they allow to transmit data through the Internet in an encrypted way, maintaining the privacy and security in the information exchange between the client application of the user and the SaaS application store in the cloud.

2.4 Technological Challenges in Cloud Computing

Cloud computing has shown to be a very effective paradigm according to its features such as on-demand self-service since the custermors are able to provision computing capabilities without requiring any human interaction; broad network access from heterogeneous client platforms; resource pooling to serve multiple consumers; rapid elasticity as the capabilities appear to be unlimited from the consumer's point of view; and a measured service allowing a pay-per-use business model. However, in order to offer such a advantageous platform, there are some weak points that are needed to take into account. Next, we present some of these issues:

- Security, privacy and confidence: Since the data can be distributed on different servers, and "out of the control" of the customer, there is a necessity of managing hardware for computation with encoding data by using robust and efficient methods. Also, in order to increase the confidence of the user, several audits and certifications of the security must be performed.

- Availability, fault tolerance and recovery: to guarantee a permanent service (24x7) with the use of redundant systems and to avoid net traffic overflow.
- Scalability: In order to adapt the necessary resources under changing demands of the user by providing an intelligent resource management, an effective monitorization can be used by identifying a priori the usage patterns and to predict the load in order to optimize the scheduling.
- Energy efficiency: It is also important to reduce the electric charge by using microprocessors with a lower energy consumption and adaptable to their use.

3 Cloud Computing for Business Intelligence Processes

As it was stated in the introduction of this work, the philosophy of Business Intelligence systems is to satisfy the necessity of analyzing large amounts of data in a short period of time, usually in just a matter of seconds or minutes. This high volume of data may come as a result of different applications such as prediction of loan concessions and credit policies to clients based on risks, classification or clustering of clients for beam marketing, product recommendations and extraction of patterns from commercial transactions among others.

In the remainder of this section, we will first present an architecture to develop BI solutions on a Cloud Computing platform. Then, we will stress the goodness on the use a Cloud Computing with respect to other similar technologies.

3.1 Organization of the Cloud Computing Environment

To address the goals stated in the beginning of this section, in [4, 8] the authors revisited a basic model and process for analyzing structured and unstructured user generated content in a business warehouse. This data management organization architecture based on clouds follows a four layer architecture, although there exists several similar approaches [21, 23]. We must point out that the three first tiers are common for DM and BI approaches, whereas the last one is specifically designed for data warehousing and On-Line Analytical Processing (OLAP) applications. The description of these components is enumerated below:

- The first level is the infrastructure tier based on Cloud Computing. It follows the structure introduced in Section 2, that is, it includes a network architecture with many loosely coupled computer nodes for providing a good scalability and a fault tolerance scheme. As suggested, the system must take into account a dynamic/elastic scheme such that the performance, cost and energy consumption of the node machines is managed at run-time.
- The second layer is devoted to the parallel data storage. The relational data bases may have difficulties when processing the data along a big number of servers, so that there is a necessity of supporting new data base management systems based on the storage and retrieval of key/value pairs (as opposed to the relational model

based on foreign-key/primary-key relationships). Some examples of systems that are optimized for this purpose are Google BigTable [5] or Sector [13].

- The third level is the execution environment. Due to the large number of nodes, Cloud Computing is especially applicable for distributed computing tasks working on elementary operations. The most known example of cloud computing execution environment is probably Google MapReduce [7] and its open source version Hadoop [10], but other projects can be found as feasible alternatives [17, 33]. All these environments aim at providing elasticity by allowing to adjust resources according to the application, handling errors transparently and ensuring the scalability of the system.
- The last tier is the high querying language tier, which is oriented towards OLAP, and Query/Reporting Data-Warehousing tools. This layer is the interface to the user and it provides the transparency to the other tiers of the architecture. Some query languages have been proposed like Map-Reduce-Merge [36] or the Pig Latin language [25] which has been designed to propose a trade-off between the declarative style of SQL, and the low-level, procedural style of MapReduce.

3.2 On the Suitability of Cloud Computing for Business Intelligence and Data Mining

Traditionally, when having a high amount of data to be processed in a short period of time, a grid computing environment was the most suitable solution in order to reduce computational costs and to increase the flexibility of the system. The similarities with Cloud Computing are evident, since both of them are composed of loosely coupled, heterogeneous, and geographically dispersed nodes. However, the main difference between the two lies in the way the tasks are computed in each respective environment. In a computational grid, one large job is divided into many small portions and executed on multiple machines, offering a similar facility for computing power. On the other hand, the computing cloud is intended to allow the user to obtain various services without investing in the underlying architecture and therefore is not so restrictive and can offer many different services, from web hosting, right down to word processing [3].

Additionally, the advantages of this new computational paradigm with respect to other competing technologies are clear. First, Cloud application providers strive to give the same or better service and performance as if the software programs were installed locally on end-user computers, so the users do not need to spend money buying a complete hardware equipment for the software to be used, i.e. a simple PDA device is enough to run the programs on the Cloud.

Second, this type of environment for the data storage and the computing schemes allows enterprizes to get their applications up and running faster, with a lower necessity of maintenance from the IT department since it automatically manages the business demand by assigning more or less IT resources (servers, storage and/or networking) depending on the computational load in real time [30]. Finally, this

inherent elasticity of this system makes the billing of the infrastructure to be done according to the former fact.

4 Adaptation of Global Data Mining Tasks within a Cloud Computing Environment

In this section we aim at pointing out the promising future that is foreseen for the Cloud Computing paradigm regarding the implementation of DM algorithms in order to deal with very large data-sets for which it has been prohibitively expensive until this moment.

The idea behind all the proposals we will introduce is always distributing the execution of the data among all the nodes of the cloud and to transfer the least volume of information as possible to make the applications highly scalable and efficient, but always maintaining the integrity and privacy of the data [14, 29].

As introduced in Section 3, it is necessary to transform the data stored into multidimensional arrays to "Pig data" [25] to be able to carry out an online analysis process by using the MapReduce/Hadoop scheme or related approaches (such as Sector/Sphere [13]), also reducing the storage costs [8]. In this programming model, users specify the computation in terms of a map and a reduce function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks.

Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the reduce function. The reduce function accepts an intermediate key I and a set of values for that key. It merges these values together to form a possibly smaller set of values. The intermediate values are supplied to the user's reduce function via an iterator. This allows us to handle lists of values that are too large to fit in memory.

The map invocations are distributed across multiple machines by automatically partitioning the input data into a set of M splits. The input splits can be processed in parallel by different machines. Reduce invocations are distributed by partitioning the intermediate key space into R pieces using a partitioning function (e.g., hash(key) mod R). The number of partitions (R) and the partitioning function are specified by the user (see Figure 2).

In brief, these systems are oriented to distribute the data sets along the cloud and to distribute the computation among the clusters, i.e. instead of moving the data among the machines, they define mapping functions to create intermediary tuples of <key, value> and the use of reduction functions for this special processing. As an example, in a program aimed at counting the number of occurrences of each word in a large collection of documents, the map function will emit each word plus an associated count of occurrences whereas the reduce function sums together all counts emitted for a particular word.

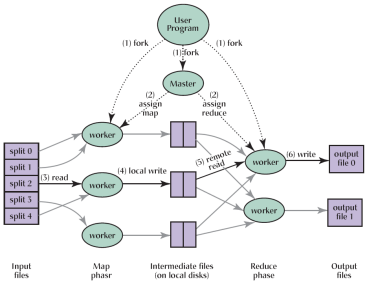

Fig. 2 Overall flow of execution a MapReduce operation

Although this is a relatively new framework, many DM algorithms have been already implemented following the guidelines of this programming model. For example, in [6] the authors present a classification model which tries to find an intermediate model between Bayes and K-nearest neighbor techniques by using a certain kind of subtree to represent each cluster which is obtained by clustering on training set by applying a minimum spanning tree MapReduce implementation, and then perform the classification using idea similar to KNN.

Another approach that is developed using the MapReduce model aims at addressing the progressive sequential pattern mining problem [15], which intrinsically suffers from the scalability problem. Two Map/Reduce jobs are designed; the candidate computing job computes candidate sequential patterns of all sequences and updates the summary of each sequence for the future computation. Then, using all candidate sequential patterns as input data, the support assembling job accumulates the occurrence frequencies of candidate sequential patterns in the current period of interest and reports frequent sequential patterns to users.

Gao et al. introduces in [12] an experimental analysis using a Random Decision Tree algorithm under a cloud computing environment by considering two different schemes in order to implement the parallelization of the learning stage. The first approach was that each node built up one or more classifiers with its local data concurrently and all classifiers are reported to a central node. Then the central node will use all classifiers together to do predictions. The second option was that each node works on a subtask of one or more classifiers and reports its result to a central node, then the central node combines work from all local nodes to generate the final classifiers and use them for prediction.

Other works are based on different cloud computing environments, a Particle Swarm Optimization was designed for the Amazon Elastic Compute Cloud (`http://aws.amazon.com`), where the candidate solutions are presented by the set of task-service pairs, having each particle to learn from different exemplars, but to learn the other feasible pairs for different dimensions. The constructive position building procedure guarantees each position was shown to be feasible and this scheme greatly reduces the search space and enhances the algorithm performance [35].

Lin and Luo proposed a novel DM method named FD-Mine [20] that is able to efficiently utilize the cloud nodes to fast discover frequent patterns in cloud computing environments with data privacy preserved. Through empirical evaluations on various simulation conditions, the proposed FD-Mine showed to deliver excellent performance in terms of scalability and execution time.

Finally, it is important to point out that there exist open source DM libraries from which the users can use the techniques implemented under these software platform. We may stress the Mahout library (`http://mahout.apache.org/`) which is mainly based on clustering approaches, but also parallel frequent pattern mining; and the Pentaho Business Analytics (`http://www.pentaho.com/big-data/`) which offers unmatched native support for the most popular big data sources including Hadoop, NoSQL (not relational data base models) and analytic databases. Additionally, the relevance in this area can be stressed by the vast amount of available commercial SaaS products such as Actuate (`http://www.actuate.com/`), ComSCI (`http://www.comsci.com/`) or FPX (`http://www.fpx.com/`); however, for most of them it is a bit unclear what they truly offer to the user in terms of which techniques they may implement for managing and mining the data, i.e. what they include in their toolbox to enable analysts to create reports and custom analyzes.

5 Concluding Remarks

In this work we have presented an overview on BI and DM applications within the Cloud Computing environment. In particular, we aimed at stressing the significance and great possibilities of this topic in the near future, since it offers an scalable framework for those high dimensional problems which are hard to overcome with the standard technologies.

Taking this into account, we have first introduced the main features of the Cloud Computing infrastructure, i.e. the different levels of abstraction which allow us to understand its nature. Then, we described its advantages with respect to other classical technologies such as grid computing and therefore how DM applications obtains higher benefits from this scheme.

Next, we have described the specific architecture that is needed in order to develop BI and DM applications within a cloud framework. Specifically, a four layer structure is suggested following the recommendations given in the specialized literature. This structure is composed of a cloud infrastructure to provide the service nodes and the communication network at the lowest level, a parallel data storage to distribute the information across the cloud, a execution environment which must take advantage of the characteristics offered by the cloud and finally the use of a high query language to fully exploit the features of the parallel execution tier.

To end with, we have presented several DM algorithms that have been migrated to Cloud Computing, examining the particularities of their implementation which are mainly based on the MapReduce/Hadoop scheme, which is currently the most important execution environment to allow an efficient parallelization of the data processing within the cloud.

References

1. Abelló, A., Romero, O.: Service-Oriented Business Intelligence. In: Aufaure, M.-A., Zimányi, E. (eds.) eBISS 2011. LNBIP, vol. 96, pp. 156–185. Springer, Heidelberg (2012)
2. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications. Springer, Heidelberg (2004)
3. Buyya, R., Yeo, C., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 25(6), 599–616 (2009)
4. Castellanos, M., Dayal, U., Sellis, T., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C. (eds.): Optimization Techniques 1974. LNBIP, vol. 27. Springer, Berlin (1975)
5. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: A distributed storage system for structured data. ACM Trans. Comput. Syst. 26(2) (2008)
6. Chang, J., Luo, J., Huang, J.Z., Feng, S., Fan, J.: Minimum spanning tree based classification model for massive data with mapreduce implementation. In: Fan, W., Hsu, W., Webb, G.I., Liu, B., Zhang, C., Gunopulos, D., Wu, X. (eds.) ICDM Workshops, pp. 129–137. IEEE Computer Society (2010)
7. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)
8. d'Orazio, L., Bimonte, S.: Multidimensional Arrays for Warehousing Data on Clouds. In: Hameurlain, A., Morvan, F., Tjoa, A.M. (eds.) Globe 2010. LNCS, vol. 6265, pp. 26–37. Springer, Heidelberg (2010)
9. Duer, W.: CRM, Customer Relationship Management. MP editions (2003)
10. Foundation, T.A.S.: Hadoop, an open source implementing of mapreduce and GFS (2012), `http://hadoop.apache.org`
11. Furht, B., Escalante, A. (eds.): Handbook of Cloud Computing. Springer, US (2010)
12. Gao, W., Grossman, R.L., Yu, P.S., Gu, Y.: Why naive ensembles do not work in cloud computing. In: Saygin, Y., Yu, J.X., Kargupta, H., Wang, W., Ranka, S., Yu, P.S., Wu, X. (eds.) ICDM Workshops, pp. 282–289. IEEE Computer Society (2009)
13. Grossman, R.L., Gu, Y., Sabala, M., Zhang, W.: Compute and storage clouds using wide area high performance networks. Future Generation Comp. Syst. 25(2), 179–183 (2009)
14. Gupta, V., Saxena, A.: Privacy Layer for Business Intelligence. In: Meghanathan, N., Boumerdassi, S., Chaki, N., Nagamalai, D. (eds.) CNSA 2010. CCIS, vol. 89, pp. 323–330. Springer, Heidelberg (2010)
15. Huang, J.-W., Lin, S.-C., Chen, M.-S.: DPSP: Distributed Progressive Sequential Pattern Mining on the Cloud. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part II. LNCS, vol. 6119, pp. 27–34. Springer, Heidelberg (2010)
16. Hurwitz, J., Bloor, R., Kaufman, M., Halper, F.: Cloud Computing for Dummies. Wiley (2010)
17. Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D.: Dryad: Distributed Data-parallel Programs from Sequential Building Blocks. In: 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems, EuroSys 2007, pp. 59–72. ACM, New York (2007)
18. Jun, L., Jun, W.: Cloud computing based solution to decision making. Procedia Engineering 15, 1822–1826 (2011)
19. Lai, C.F., Chang, J.H., Hu, C.C., Huang, Y.M., Chao, H.C.: Cprs: A cloud-based program recommendation system for digital tv platforms. Future Generation Comp. Syst. 27(6), 823–835 (2011)

20. Lin, K.W., Luo, Y.C.: A fast parallel algorithm for discovering frequent patterns. In: GrC, pp. 398–403. IEEE (2009)
21. Liyang, T., Zhiwei, N., Zhangjun, W., Li, W.: A conceptual framework for business intelligence as a service (saas bi). In: Fourth International Conference on Intelligent Computation Technology and Automation, ICICTA 2011, pp. 1025–1028. IEEE Computer Society, Washington, DC (2011)
22. McNabb, A.W., Monson, C.K., Seppi, K.D.: Parallel pso using mapreduce. In: IEEE Congress on Evolutionary Computation, pp. 7–14. IEEE (2007)
23. Mircea, M., Ghilic-Micu, B., Stoica, M.: Combining business intelligence with cloud computing to delivery agility in actual economy. Journal of Economic Computation and Economic Cybernetics Studies (in press, 2012)
24. Negash, S., Gray, P.: Business intelligence. Communications of the Association for Information Systems 13, 177–195 (2004)
25. Olston, C., Reed, B., Srivastava, U., Kumar, R., Tomkins, A.: Pig latin: a not-so-foreign language for data processing. In: SIGMOD 2008: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1099–1110. ACM (2008)
26. Ouf, S., Nasr, M.: Business intelligence in the cloud. In: IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011, pp. 650–655 (2011)
27. Papazoglou, M., Van Den Heuvel, W.J.: Service oriented architectures: Approaches, technologies and research issues. VLDB Journal 16(3), 389–415 (2007)
28. Shen, C.P., Jigjidsuren, C., Dorjgochoo, S., Chen, C.H., Chen, W.H., Hsu, C.K., Wu, J.M., Hsueh, C.W., Lai, M.S., Tan, C.T., Altangerel, E., Lai, F.: A data-mining framework for transnational healthcare system. Journal of Medical Systems, 1–11 (2011)
29. Upmanyu, M., Namboodiri, A.M., Srinathan, K., Jawahar, C.V.: Efficient Privacy Preserving K-Means Clustering. In: Chen, H., Chau, M., Li, S.-h., Urs, S., Srinivasa, S., Wang, G.A. (eds.) PAISI 2010. LNCS, vol. 6122, pp. 154–166. Springer, Heidelberg (2010)
30. Velte, A.T., Velte, T.J., Elsenpeter, R. (eds.): Cloud Computing: A Practical Approach. McGraw Hill (2010)
31. Verma, A., Llor, X., Goldberg, D.E., Campbell, R.H.: Scaling genetic algorithms using mapreduce. In: ISDA, pp. 13–18. IEEE Computer Society (2009)
32. Wang, J., Liu, Z.: Parallel data mining optimal algorithm of virtual cluster. In: Ma, J., Yin, Y., Yu, J., Zhou, S. (eds.) FSKD (5), pp. 358–362. IEEE Computer Society (2008)
33. Warneke, D., Kao, O.: Exploiting dynamic resource allocation for efficient parallel data processing in the cloud. IEEE Transactions on Parallel Distributed Systems 22(6), 985–997 (2011)
34. White, B., Yeh, T., Lin, J., Davis, L.: Web-scale computer vision using mapreduce for multimedia data mining. In: Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD 2010, pp. 9:1–9:10. ACM, New York (2010)
35. Wu, Z., Ni, Z., Gu, L., Liu, X.: A revised discrete particle swarm optimization for cloud workflow scheduling. In: Liu, M., Wang, Y., Guo, P. (eds.) CIS, pp. 184–188. IEEE (2010)
36. Yang, H., Dasdan, A., Hsiao, R.L., Parker, D.S.: Map-reduce-merge: simplified relational data processing on large clusters. In: SIGMOD 2007: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pp. 1029–1040. ACM (2007)