Performance evaluation of large-scale Information Retrieval systems scaling down

Position paper

Fidel Cacheda, Víctor Carneiro, Diego Fernández, Vreixo Formoso
Facultad de Informática
Campus de Elviña s/n
15071, A Coruña, Spain
{fidel.cacheda, victor.carneiro, dfernandezi, vformoso}@udc.es

ABSTRACT

The performance evaluation of an IR system is a key point in the development of any search engine, and specially in the Web. In order to get the performance we are used to, Web search engines are based on large-scale distributed systems and to optimise its performance is an important aspect in the literature.

The main methods, that can be found in the literature, to analyse the performance of a distributed IR system are: the use of an analytical model, a simulation model and a real search engine. When using an analytical or simulation model some details could be missing and this will produce some differences between the real and estimated performance. When using a real system, the results obtained will be more precise but the resources required to build a large-scale search engine are excessive.

In this paper we propose to study the performance by building a scaled-down version of a search engine using virtualization tools to create a realistic distributed system. Scaling-down a distributed IR system will maintain the behaviour of the whole system and, at the same time, the computer requirements will be softened. This allows the use of virtualization tools to build a large-scale distributed system using just a small cluster of computers.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*

General Terms

Information Retrieval

Keywords

Distributed Information Retrieval, Performance evaluation, Scalability

1. INTRODUCTION

Web search engines have changed our perspective of the search process because now we consider normal being able to search through billions of documents in less than a second. For example, we may not quite understand why we have to wait so long in our council for a certificate as they just have to search through a "few" thousand/million records.

However, Web search engines have to use a lot of computational power to get the performance we are used to. This computational power can only be achieved using large-scale distributed architectures. Therefore, it is extremely important to determine the distributed architectures and techniques that allow clear improvements in the system performance.

The performance of a Web search engine is determined basically by two factors:

- Response time: the time it takes to answer the query. This time includes the network transfer times that, in Internet, will take a few hundred milliseconds; and the processing time in the search engine, that is usually limited to 100 milliseconds.
- Throughput: the number of queries the search engine is able to process per second. This measure usually has to maintain a constant ratio, but also deals with peak loads.

From the user's point of view, only the response time is visible and it is the main factor, keeping constant the quality of the results: the faster the search engine is able to answer the better. From the search engine point of view both measures are important. Once an upper limit has been set up for the response time (e.g. a query should be answered in less than 100 milliseconds), the objective is to maximise the throughput.

From the search engine point of view another two factors have to be taken into account:

- Size: the number of documents indexed by the search engine. Not so long ago, Google published in its main page the number of Web pages indexed. Nowadays, the main commercial search engines do not make public detailed figures, although the estimations are in the order of 20 billion documents.

- Resources: the number of computers used by the search engine. This could be considered from the economical perspective as the cost of the distributed system. In [2] Baeza-Yates et al. estimate that, a search engine will need about 30 thousand computers to index 20 billion documents and obtain a good performance.

If we want to compare different distributed indexing models or test some new techniques for a distributed search engine (e.g. a new cache policy), usually we will fix the size of the collection and the resources and then, measure the performance in terms of response time and throughput.

Ideally, we would need a replica of a large-scale IR system (for example, one petabyte of data and one thousand computers) to measure performance. However, this would be extremely expensive and no research group, or even commercial search engine, can devote such amount of resources only to evaluation purposes.

In this article we present a new approach for performance evaluation of large-scale IR systems based on scaling-down. We consider that, creating a scaled-down version of an IR system, will produce valid results for the performance analysis, using very few resources. This is an important point for commercial search engines (from the economical point of view), but it is more important for the research groups because this could open the experimentation on large-scale IR to nearly any group.

The rest of the paper is organised as follows. In Section 2 we present the main approaches for performance evaluation. Section 3 analyses our proposal and Section 4 concludes this work and describes some ideas for future works.

2. PERFORMANCE EVALUATION

In the literature there are many articles that evaluate the performance of a search engine or one of its components. We do not intend to present an exhaustive list of papers about performance evaluation but to present the main methods used by the researchers, specially of a large-scale IR system.

The main methods to test the performance of a search engine are the following:

- An analytical model.
- A simulation model.
- A real system or part of a real system.

A clear example of study based on an analytical model can be found in [6]. In this work, Chowdhury et al. use the queueing network theory to model a distributed search engine. The authors model that the processing time in a query server is a function of the number of documents indexed. They build a framework in order to analyse distributed architectures for search engines in terms of response time, throughput and utilization. To show the utility of this framework, they provide a set of requirements and study different scalability strategies.

There are many works based on simulation that study the performance of a distributed IR system. [3] is one of the first. In this work, Burkowski uses a simple simulation model to estimate the response time of a distributed search engine, and uses one server to estimate the values for the simulation model (e.g. the reading time from disk is approximated as a Normal distribution). Then, the simulation model represents a clusters of servers and estimates the response times using local index organisation (named uniform distribution). However, the network times are not considered in the simulation model.

Tomasic and Garcia-Molina [12] also used a simulation model to study the performance of several parallel query processing strategies using various options for the organization of the inverted index. They use different simulation models to represent the collection documents, the queries, the answer set and the inverted lists.

Cacheda et al. in [4] include also a network model to simulate the behaviour of the network in a distributed IR system. They compare different distribution architectures (global and local indexing), identify the main problems and present some specific solutions, such as, the use of partial result sets or the hierarchical distribution for the brokers.

Other authors use a combination of both approaches. For example, in [10], Ribeiro-Neto and Barbosa use a simple analytical model to estimate the processing time in a distributed system. This analytical model calculates the seek time for a disk, the reading time from disk of an inverted list, the time to compare and swap two terms and the transfer time from one computer to another. In their work, they include a small simulator to represent the interference among the various queries in a distributed environment. They compare the performance of a global index and a local index and study the effect of the network and disk speed.

Some examples of works experimenting with a real IR system could be [1] or [9]. In the first work, Badue et al. study the imbalance of the workload in a distributed search engine. They use a configuration of 7 index servers and one broker to index a collection of 10 million Web pages. In their work, the use of a real system for testing was important to detect some important factors for imbalance in the index servers. They state that the correlations between the term frequency in a query log and the size of its inverted list lead to imbalances in query execution times, because these correlations affect the behaviour of the disk caching.

Moffat et al. in [9] study a distributed indexing technique named *pipelined distribution*. In a system of 8 servers and one broker, they index the TREC Terabyte collection [7] to run their experiments. The authors compare three distributed architectures: local indexing (or document partitioning), global indexing (or term partitioning) and pipelining. In their experiments the pipelined distribution outperforms the term partitioning, but not the document partitioning due to a poor workload balancing. However, they also detect some advantages over the document distribution: a better use of memory and fewer disk seeks and transfers.

The main drawback for an analytical model is that it cannot represent all the characteristics of a real IR system. Some features have to be dropped to keep the model simple and easy to implement.

Using a simulation model, we can represent more complex behaviours than an analytical model. For example, instead of assuming a fixed transfer time for the network, we can simulate the behaviour of the network (e.g. we could detect a network saturation). But, again, not all the features of a real system could be implemented. Otherwise, we will end up with a real IR system and not a simulation model.

In both cases, it is important to use a real system to estimate the initial values of the model (analytical or simulated) and, in fact, this is a common practise in all the research works. In a second step, it is also common to compare the results of the model with the response obtained from a real system, using a different configuration, in order to validate the model.

However, when the models are used to extrapolate the behaviour of a distributed IR system, for example increasing the number computers, the results obtained may introduce a bigger error than expected. For example, a simulation model of one computer, when compared with a real system, has an accuracy of 99% (or an error of 1%). But, what is the expected error when simulating a system with 10 computers: 1% or 10%?

This problem is solved by using a real system for the performance evaluation. But, in this case, the experiments will be limited by the resources available in the research groups. In fact, many researchers run their experiments using 10-20 computers. Considering the size of data collections and the size of commercial search engines, this could not be enough to provide interesting results for the research community. In this sense, the analytical or simulation models allow us to go further, at the risk of increasing the error ratio of our estimations.

3. OUR PROPOSAL

In this article we propose to use a scaled-down version of a search engine to analyse the performance of a large-scale search engine.

Scaling down has been successfully applied in many other disciplines, and it is specially interesting when the development of a real system is extremely expensive. For example, in the shipping industry the use of scale models in basins is an important way to quantify and demonstrate the behaviour of a ship or structure, before building a real ship [5].

The use of a wind tunnel is also quite common in the aeronautical or car industries. Specially in the former, the scaled-down models of planes or parts of a plane are important to analyse the performance of the structure [13]. Also in architecture scaled-down models are used to test and improve the efficiency of a building [8].

In the world of search engines, is it possible to build a scaled-down version of a search engine?

Let us say that we want to study the performance of a large-scale search engine, composed of 1000 computers, with the following parameters:

- The size of the collection is 1 petabyte.
- Each computer has a memory of 10 gigabytes.
- Each computer has a disk of 1 terabyte.
- The computers are interconnected using a high speed network (10Gbits/second).

From our point of view, maintaining the 1000 computers as the core of the distributed system, if we apply a scale factor of 1:1000 we will have a scaled-down version of the search engine with the following parameters:

- The size of the collection is 1 gigabyte.
- Each computer has a memory of 10 megabytes.
- Each computer has a disk of 1 gigabyte.
- The computers are interconnected using a high speed network (10Mbits/second).

One important point is that the scale factor does not apply to the number of computers. The computers constitute the core of the distributed system and therefore cannot be diminished, instead they are scaled-down. This is equivalent to build a scaled-down version of a building: the beams are scaled-down but not diminished.

In this way, we expect to obtain a smaller version of the large-scale search engine, but with the same drawbacks and benefits.

The next step is how to build this scaled-down version of a search engine.

The first and trivial solution is to use 1000 computers with the requirements stated previously. These would be very basic computers nowadays but, anyway, it could be quite complicated to obtain 1000 computers for a typical research group. It could be a little bit easier for a commercial search engine if they could have access to obsolete computers from previous distributed systems, but it is not straightforward.

A more interesting solution would be to use virtualization tools to create the cluster of computers. Virtualization is a technology that uses computing resources to present one or many operating systems to user. This technology is based on methodologies like hardware and software partitioning, partial or complete machine simulation and others [11]. In this work, we are interested in the virtualization at the hardware abstraction layer to emulate a personal computer. Some well-known commercial PC emulators are KVM[1], VMware[2], VirtualBox[3] or Virtual PC[4].

With this technology it could be possible to virtualize a group of scaled-down computers using just one real computer. In this way, with a small cluster of computers (e.g. 20 computers) we could virtualize the whole scaled-down search engine.

For example, using a cluster of 20 computers, will require that each computer virtualizes 50 computers with 10 megabytes of memory and 1 gigabyte of disk. Roughly speaking, this would take half a gigabyte of memory and 50 gigabytes of disk from the real machine, which should be easily handled by any modern computer.

To the best of our knowledge, all the virtualization tools allow you to set the size of memory and disk for the virtualized host. Also, some of them (e.g. VMware) can set a limit for the network usage, in terms of average or peak, and for the CPU speed.

From our point of view, these requirements should be enough to scale-down a host of a distributed search engine. Some other parameters could also be considered when scaling down a search engine, such as, disk or memory speed. We are aware of some solutions in this sense[5], but these low level parameters could be quite hard to virtualize. However, we doubt about the usefulness of these parameters in the performance analysis, while the performance of the real computer is not degraded. In any case, in future works it

[1] http://www.linux-kvm.org/

[2] http://www.vmware.com/

[3] http://www.virtualbox.org/

[4] http://www.microsoft.com/windows/virtual-pc/

[5] http://sourceforge.net/apps/trac/ioband/

would be interesting to study in depth the effect of these low level parameters in the performance analysis of a scaled-down search engine.

The use of virtualization is not only interesting to reduce the resources required to build a scaled-down version of a search engine. It could be very appealing to test the effect of new technologies in the performance of a large-scale search engine. For example, let us say that we want to compare the performance of the new SSD (Solid State Drive) memories versus the traditional hard drives. To build a whole search engine using SSD memories would be a waste of resources until the performance has been tested. But, buying a cluster of computers with SSD memory and building a scaled-down version of the search engine is feasible and not very expensive. In this way, we could test and compare the performance of this new technology.

4. CONCLUSIONS

This paper presents a new approach to the performance evaluation of large-scale search engines based on a scaled-down version of the distributed system.

The main problem, when using an analytical or simulation model for evaluation purposes, is that some (important) details could be missing in order to make the model feasible and so, the estimations obtained could differ substantially from the real values.

If we use a real search engine for performance evaluation, the results obtained will be more precise but will depend on the resources available. A distributed system composed of a few computers does not constitute a large-scale search engine and the resources required to build a representative search engine are excessive for most researchers.

We suggest to build a scaled-down version of a search engine using virtualization tools to create a realistic cluster of computers. By using a scaled-down version of a computer we expect to maintain the behaviour of the whole distributed system at the same time that the hardware requirements are softened. This would be the key to use the virtualization tools to build a large distributed system using a small cluster of computers.

This research is at an early stage, but we strongly believe that this would be a valid technique to analyse the performance of a large-scale distributed IR system.

In the near future we plan to develop a scaled-down search engine using a small cluster of computers. We would like to compare the performance of the scaled-down search engine with an equivalent real search engine to test the accuracy of this methodology.

5. ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Education and Science under project TIN2009-14203.

6. REFERENCES

[1] C. S. Badue, R. Baeza-Yates, B. Ribeiro-Neto, A. Ziviani, and N. Ziviani. Analyzing imbalance among homogeneous index servers in a web search system. *Inf. Process. Manage.*, 43(3):592–608, 2007.

[2] R. A. Baeza-Yates, C. Castillo, F. Junqueira, V. Plachouras, and F. Silvestri. Challenges on distributed web retrieval. In *ICDE*, pages 6–20, 2007.

[3] F. J. Burkowski. Retrieval performance of a distributed text database utilizing a parallel processor document server. In *DPDS '90: Proceedings of the second international symposium on Databases in parallel and distributed systems*, pages 71–79, New York, NY, USA, 1990. ACM.

[4] F. Cacheda, V. Carneiro, V. Plachouras, and I. Ounis. Performance analysis of distributed information retrieval architectures using an improved network simulation model. *Inf. Process. Manage.*, 43(1):204–224, 2007.

[5] J.-H. Chen and C.-C. Chang. A moving piv system for ship model test in a towing tank. *Ocean Engineering*, 33(14-15):2025 – 2046, 2006.

[6] A. Chowdhury and G. Pass. Operational requirements for scalable search systems. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 435–442, New York, NY, USA, 2003. ACM.

[7] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2004 terabyte track. In *TREC*, 2004.

[8] J.-H. Kang and S.-J. Lee. Improvement of natural ventilation in a large factory building using a louver ventilator. *Building and Environment*, 43(12):2132 – 2141, 2008.

[9] A. Moffat, W. Webber, J. Zobel, and R. A. Baeza-Yates. A pipelined architecture for distributed text query evaluation. *Inf. Retr.*, 10(3):205–231, 2007.

[10] B. Ribeiro-Neto and R. A. Barbosa. Query performance for tightly coupled distributed digital libraries. In *Proceedings of the third ACM conference on Digital libraries*, pages 182–190, Pittsburgh, Pennsylvania, United States, 1998. ACM Press.

[11] N. Susanta and C. Tzi-Cker. A survey on virtualization technologies. Technical report.

[12] A. Tomasic and H. Garcia-Molina. Query processing and inverted indices in shared: nothing text document information retrieval systems. *The VLDB Journal*, 2(3):243–276, 1993.

[13] S. Zhong, M. Jabbal, H. Tang, L. Garcillan, F. Guo, N. Wood, and C. Warsop. Towards the design of synthetic-jet actuators for full-scale flight conditions. *Flow, Turbulence and Combustion*, 78(3):283 – 307, 2007.