Performance Analysis of CORBA-based Distributed Data Handling Systems

Aleksey Burdakov
post-graduate student
at the BMSTU,
2000

Abstract

The process of development of distributed data handling systems (DDHS) requires the future system performance analysis. Existing performance characteristics evaluation methods, which are mostly based on the Queuing Theory (QT), have a set of disadvantages making it difficult to use them on the earliest stages of a project. This paper surveys existing methods of analysis and makes some proposals of new ones. The directions of the future research are given as well.

1. Introduction

The process of development of distributed data handling systems (DDHS) requires the future system performance analysis. Existing performance characteristic evaluation methods, which are mostly based on the Queuing Theory (QT), have a set of disadvantages making it difficult to use them on the earliest stages of a project.

Nowadays, it becomes more usual to use special software (so-called "middle-ware"), which helps to build complex distributed systems on its base. Such software "glues" different parts of a system into one piece. CORBA is believed to be the maturest solution in this area, which allows building of DDHS in heterogeneous environments, using object-oriented technologies. Database management system (DBMS) – is an essential component of any DDHS.

In this article an overview of CORBA-based DDHS development methods, as well as several proposals of systems and their components (relational RDBMS, object-relational ORDBMS and object-oriented OODBMS) performance evaluation methods are given.

2. CORBA-based DDHS Performance Evaluation

CORBA (Common Object Request Broker Architecture) allows developing of distributed systems based on heterogeneous components (objects) by tiding them together, giving a wide range of services: naming, transactional, persistence services, etc. ORB (Object Request Broker) supports seamless intercommunication between remote objects gives independence of physical location, platform features (software and hardware) and the language a system was designed on. By means of IIOP, ORB also gives an independence of particular ORB supplier. The features mentioned above make CORBA a universal layer, which allows building of DDHS from different components, i.e. DBMS, clients, application servers, etc.

2.1. Performance evaluation

Intercommunication between CORBA objects is performed via remote object method calls, and remote objects activation on remote servers (the notion "remote" is used here for general case, i.e. sometimes an object can reside on the same node and be "local").

A call of a remote object includes the following steps:

1. marshaling of data into CDR (Common Data Representation) format,
2. transporting of a query through an established TCP connection,
3. server skeleton identification via RefID or object_key,
4. unmarshaling and method processing,
5. marshaling of the result, and transporting it back to the caller.

An object activation process includes the following steps:

1. looking up of a "locator" file for the object address, and sending a request through TCP protocol to a server,
2. server starting, object creation, socket point-of-call creation in BOA (Basic Object Adapter), setting a unique ID, IOR (Interoperable Object Reference) creation,
3. object reference returning to the caller, and establishing a socket connection via TCP with the server.

2.2. CORBA and OO/ORDBMS integration methods

DBMS is the major component of any distributed data handling system. There are a few ways of CORBA and OO/ORDBMS integration:

1. via DBMS API (only static API is presented as object methods),
2. at the level of database objects:
    a. via POS (Persistence Object Service),
    b. via ODA (Object Database Adapter).

In the first case, the query execution time is approximately equal to the sum of query execution in DBMS and query transportation time through ORB.

In the second case, the query execution time depends on various factors, among them: presence of an object in an object cache, object search in cache, etc. It is necessary to know the object cache hit probability and object management time in the cache. In this case the total execution time is equal to data transportation time through ORB, data processing time in a DBMS and object storage and search time in a cache.

3. DBMS Performance Evaluation

The main part of processing job in a DDHS is done on DBMS servers, where usually huge tables are processed into small result sets, and then transferred to the calling client as a result for its query. This makes it quite important to analyze performance of this part of a DDHS.

There are three major DBMS data models: hierarchical (IMS), network (CODASYL) and relational. Although relational model is semantically weak, almost all modern DBMSs are based on relational, object-relational and object-oriented models. The latter DBMS type is the most perspective. As it is stated in a number of publications, the situation will remain motionless in the nearest future with a little increase of OODBMS part.

The data processing in the DBMS types mentioned above is performed by means of non-procedural (declarative) language SQL (Structured Query Language) and its variations. SQL was proposed soon after the Codd's publication on relational model in 1970. Since then SQL has become both de-facto and de-jury standard for relational and even for non-relational DBMSs. There are two major standards nowadays: SQL-1999 by ANSI and ISO committees for ORDBMS and ODMG 3.0 proposed by ODMG (Object Data Management Group).

3.1. DBMS Query Processing and its Performance Evaluation

Having knowledge of query processing principles, statistics on data in a database, hardware and software platform performance characteristics it is possible to predict query execution time. However it is necessary to perform all the query optimization stages to get the final query execution plan, which is executed by a DBMS. A lot of researches have been carried out in the field of RDBMS, ORDBMS and OODBMS query optimization recently. In spite of differences, they all are based on algorithms and principles described by Codd in 1972. These algorithms are based on relational algebra, by means of which it is possible to transform the form of a query to another with the minimal execution time.

Query execution involves the following steps: query parsing and validation, view resolution, query optimization, plan compilation, and query execution. The result execution plan is mostly influenced by optimization phase. Optimization includes the following two steps: heuristic optimization and cost-based optimization.

Heuristic (or algebraic) optimization is a query transformation independently of system-dependent cost model. Such an optimization is based on the rules (e.g. bringing selections into productions), that are used to transform a query into more efficient form.

Cost-based optimization differs from heuristic optimization, since it is based on specific knowledge on a database and DBMS (i.e. query execution methods, physical storage structures, access methods, indices, etc.).

In the work, a mathematical method of query execution time evaluation in relational DBMS (based on the knowledge of RDBMS query optimization methods, conceptual scheme parameters, and statistics on a database) is proposed.

3.2. Query Processing Features in OO/ORDBMS

Query processing in OODBMS and ORDBMS differs from RDBMS due to the following features of the former:

1. support of complex logical data structures, e.g. nested objects and its sets: lists, sets, bags and arrays,
2. special storage methods, which are different from plain relational tables in case of RDBMS, e.g. clustering, storing of nested objects with parent records, etc.,
3. query language (SQL 1999 and OQL 3.0) includes nested queries in all causes of queries (SELECT, FROM and WHERE),
4. query language includes special "path expressions", those help to access nested objects,
5. special index structures for nested objects.

4. Conclusion

In the framework of this research it is planned to develop DDHS performance evaluation methods, carry out experiments on a real information system with the help of “KISP” (Automated Systems Project Development Lifecycle and Decision Support System). It is also planned to prove the proposed methods on the basis of the experiment results.

References

1. Григорьев Ю.А., Плутенко А.Д. Жизненный цикл проектирования баз данных. – Благовещенск: Изд-во Амурского гос. ун-та, 1999. - 266 с.
2. Григорьев Ю.А. Информационная система сопровождения жизненного цикла разработки распределенных систем обработки данных // Вестник МГТУ. Сер. Приборостроение. - 1999. - № 2. - С. 37-45.
3. Григорьев Ю.А., Бурдаков А.В., Плутенко А.Д. Анализ характеристик производительности распределенных систем обработки данных // Проблемы построения и эксплуатации систем обработки информации и управления: Сборник статей. - Вып. - 1. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. – С.11-17.
4. Саймон А.Р. Стратегические технологии баз данных: менеджмент на 2000 год. - М.: Финансы и статистика, 1999. – 478 с.
5. Дэйт К. Дж. Введение в системы баз данных. - К.: Диалектика, 1998. - 784 с.
6. Дунаев С. Доступ к базам данных и техника работы в сети. Практические примеры современного программирования. - М.: ДИАЛОГ-МИФИ, 1999 - 416 с.
7. Григорьев Ю.А., Плутенко А.Д. Оценка времени выполнения запросов к реляционной системе управления базами данных. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. – 56 с.
8. Тиори Т., Фрай Дж. Проектирование структур баз данных. - М.: Мир, 1985. - 320 с.
9. Орфали Р., Харки Д., Эдвардс Д. Основы CORBA. - М.: МАЛИП, Горячая линия - Телеком, 1999. - 318с.
10. Seigel, Jon. CORBA fundamentals and programming / written and edited by Jon Seigel. - John Willey & Sons, Inc., 1996 - 693 p.
11. Hiroshi Ishikawa. Object-Oriented Database System. Design and Implementation for Advanced Applications, Springer-Verlag, Tokyo, 1993.
12. Kim W. Introduction to object-oriented databases. -Cambridge(Ma);London: The MIT press, 1990. -234 p.: ill..
13. Britts S. Object database design: Diss.. -Stockholm, 1994. -Pag.var.: ill.Report series/Stockholm univ..Department of computer and systems sciences.
14. Advances in object-oriented database systems: Proc.of the NATO advanced study inst.on object-oriented database systems,held in Izmar,Aug.6-16,1993/ Ed A.Dogas et al. -Berlin et al: Springer, 1994. -XI, 515 p.: ill NATO ASI (advanced science institutes) series. Ser.F, Computer and systems sciences; Vol.130.
15. Steenhagen H.J. Optimization of object query languages. Thesis University Twente Enschede, 1995 -207p.