Key Points

The main application of Web transactions is in eCommerce applications. In a minimal configuration, the architecture for Web transactions consists of a client, a web server and a database server. Database access is provided at the web server level, i.e., by embedding database access into Java servlets or JavaServer Pages (JSP). More sophisticated architectures consider a dedicated application server layer (i.e., an implementation of the Java Enterprise Edition specification Java EE) and support distributed databases. Communication between application and database layer is usually implemented on the basis of JDBC (Java Database Connectivity), ODBC (Open Database Connectivity), or native database protocols. Transaction support at application server level is provided by services like JTS (Java Transaction Service) via the Java Transaction API (JTA) in the Java world, or OTS (Object Transaction Service) in CORBA. Essentially, these services allow to associate several application server calls and their interactions with resource managers with the same transaction context and to coordinate them by using a two phase commit (2PC) protocol. Thus, Web transactions mostly focus on atomic commit processing. Similarly, protocols on the Web service stack exploit 2PC to atomically execute several Web services.

Multi-tier web applications require special support for failure handling at application level (application recovery). In order to increase the degree of scalability of multi-tier architectures for Web transactions, application servers can be replicated. To avoid that the database backend then becomes a bottleneck, dedicated caching strategies at the middle tier are applied. Web transactions can be part of Transactional processes.

Cross-references

- ▶ Application Server
- ▶ Caching
- ▶ Transactional Processes
- ▶ Web Service

Recommended Reading

1. Barga R., Lomet D., Shegalov G., and Weikum G. Recovery Guarantees for Internet Applications. ACM Trans. Internet Technol., 4(3):289–328, 2004.
2. Burke R. and Monson-Haefel R. Enterprise JavaBeans 3.0. O'Reilly, 2006.
3. Luo Q., Krishnamurthy S., Mohan C., Pirahesh H., Woo H., Lindsay B., and Naughton J. Middle-tier database caching for e-business. In Proc. ACM SIGMOD Int. Conf. on Management of Data, 2002, pp. 600 –611.

Web Usage Mining

▶ Data, Text, and Web Mining in Healthcare

Web Views

Alexandros Labrinidis
Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, USA

Synonyms

Web Views; HTML fragment

Definition

Web Views are web pages or web page fragments that are automatically created from base data, which are typically stored in a database management system (DBMS).

Key Points

Although caching of HTML pages (and fragments) has been proposed in the literature as early as the mid-1990s, the term Web View was introduced in 1999 [2] to denote that these fragments are generated through queries made to a back-end DBMS.

The concept of a *Web View* facilitates the materialization of dynamically generated HTML fragments outside the DBMS. The main advantage of materializing Web Views (outside the DBMS, e.g., at the web server) is that the web server need not query the DBMS for every user request, thus greatly improving query response time for the user [3]. On the other hand, for the quality of the data returned to the user to be high, the system must keep materialized Web Views fresh (in the background). This essentially decouples the processing of queries at the web server from the processing of updates at the DBMS (which are also propagated to materialized Web Views).

Materializing a Web View presents a trade-off. On the one hand, it can greatly improve response time for user-submitted queries. On the other hand, it generates a background overhead for the Web View to be kept fresh (essentially a materialization decision can be

viewed as a "*contract*" for the Web View to be kept fresh). As such, selecting which Web View to materialize is an important problem that has received attention [5,6]. This problem is essentially similar to the view selection problem in traditional DBMSs [5], with added complexity due to the online nature of the Web and the need for any solution to be highly dynamic, constantly adapting to changing conditions and workloads.

Having identified the set of Web Views to materialize, there is the issue of determining the order by which to propagate updates to them. Since one update to a base relation can trigger the refresh of multiple different materialized Web Views, the order by which these are updated can make an impact on the overall quality of data returned to the user. This is essentially a special-case online scheduling problem, which has been addressed in [4].

Web Views have been used successfully to decouple the processing of queries (to generate dynamic, database-driven web pages) from that of updates (to the content stored inside the DBMS used to driven the web site). This enables much better performance (in terms of response to user queries) without sacrificing the freshness of the data served back to the user.

Cross-references

► View Maintenance

Recommended Reading

1. Gupta H. and Mumick I.S. Selection of views to materialize under a maintenance cost constraint. In Proc. 7th Int. Conf. on Database Theory, 1999, pp. 453–470.
2. Labrinidis A. and Roussopoulos N. Alexandros On the materialization of Web views. In Proc. ACM SIGMOD Workshop on The Web and Databases, 1999, pp. 79–84.
3. Labrinidis A. and Roussopoulo N. Web View materialization. In Proc. ACM SIGMOD Int. Conf. on Management of Data, 2000, pp. 367–378.
4. Labrinidis A. and Roussopoulos N. Update propagation strategies for improving the quality of data on the Web. In Proc. 27th Int. Conf. on Very Large Data Bases, 2001, pp. 391–400.
5. Labrinidis A. and Roussopoulos N. Balancing performance and data freshness in Web database servers. In Proc. 29th Int. Conf. on Very Large Data Bases, 2003, pp. 393–404.
6. Labrinidis A. and Roussopoulos N. Exploring the tradeoff between performance and data freshness in database-driven Web servers. VLDB J., 13(3):240–255, 2004.

Web Widget

► Snippet

What-If Analysis

Stefano Rizzi
University of Bologna, Bologna, Italy

Definition

In order to be able to evaluate beforehand the impact of a strategic or tactical move so as to plan optimal strategies to reach their goals, decision makers need reliable predictive systems. What-if analysis is a data-intensive simulation whose goal is to inspect the behavior of a complex system, such as the corporate business or a part of it, under some given hypotheses called scenarios. In particular, what-if analysis measures how changes in a set of independent variables impact a set of dependent variables with reference to a given simulation model such a model is a simplified representation of the business, tuned according to the historical corporate data. In practice, formulating a scenario enables the building of a hypothetical world that the analyst can then query and navigate.

Historical Background

Though what-if analysis can be considered as a relatively recent discipline, its background is rooted at the confluence of different research areas, some of which date back decades ago.

First of all, what-if analysis lends some of the techniques developed within the simulation community, to contextualize them for *business intelligence*. Simulations are used in a wide variety of practical contexts, including physics, chemistry, biology, engineering, economics, and psychology. Much literature has been written in this field over the years, mainly regarding the design of simulation experiments and the validation of simulation models [5,8,9].

Another relevant field for what-if analysis is economics that provide the insights into business processes necessary to build and test simulation models. For instance, in [1] a set of alternative approaches to forecasting are surveyed, and useful guidelines for selecting the best ones according to the availability and reliability of knowledge are given.

Finally, what-if analysis relies heavily on database and *data warehouse* technology. Though data warehousing systems have been playing a leading role in supporting the decision process over the last decade, they are aimed at supporting analysis of past data