Winter 12-19-2001

The Extended Commands for the Management of Database Snapshots for Internet Decision Support Applications

David Chao

THE EXTENDED COMMANDS FOR THE MANAGEMENT OF DATABASE SNAPSHOTS FOR INTRANET DECISION SUPPORT APPLICATIONS

David Chao
College of Business
San Francisco State University
San Francisco, CA 94132
E-mail: dchao@sfsu.edu

ABSTRACT

Database snapshots have been an effective way of supporting decision support applications, and will continue to play an important role in providing data in an Intranet decision support environment. This paper discusses the special characteristics of the database snapshot management in an Intranet decision support environment, review the major components involved in the management of snapshots, and then define new commands for the management of snapshots in this environment.

1. INTRODUCTION

Database snapshots are read-only copies of a selected portion of the database representing the state of the database, and a user's view of the operational data at a fixed point in time (snaptime) [1]. A snapshot is initiated by a DEFINE SNAPSHOT statement that defines the snapshot contents. In Structure Query Language, SQL, this command might look like:

DEFINE SNAPSHOT <snapshot-name> AS <query>

where <query> can be any valid SQL SELECT command. Database records that satisfy the query are said to be relevant or qualified to the snapshot. Relevant records are materialized and stored under the snapshot-name. A snapshot is read-only from the user's point of view to maintain its consistency with the database at the snaptime. As updates occur to the database, the snapshot will eventually become "stale" and its value for decision making will diminish. To bring a snapshot to a new state consistent with the database, the user issues a REFRESH command. In SQL this command is:

REFRESH SNAPSHOT <snapshot-name>

Thus, snapshots correspond to "materialized views" except that they are "refreshed" only when the user explicitly issues a refresh request. Snapshots are created mainly for decision support applications. Many such applications call for historical or end-of-period data. In decision support systems users may either prefer not to use, or should not be allowed to use, real time data. Current materialized view research emphasizes using materialized views to efficiently answer queries with real time data [2]. Unlike materialized views, snapshots are not designed to provide real time data.

A decision support database (DSDB), such as a data warehouse, contains data from various sources including internal and external data. Data from these sources are first extracted, cleaned, and then integrated before loaded to the DSDB [3]. A DSDB is read-only, and holds information that is consistent with the various data sources as of a specific point in time. Updates to the DSDB will be done according to a predefined schedule. A DSDB is essentially a generalization of the snapshot concept in the sense that its content remains static between two refreshes so that users involved in decision support and analysis activities can work with a relatively static copy of the database.

An Intranet is an organization's internal network that provides an Internet-like environment within the organization for improving communication, collaboration, and information sharing. Intranets are being used as a platform for developing business applications to support managerial decision-making. Today, many companies develop Enterprise Information Portal (EIP) to integrate Intranet applications. Decision support databases and tools such as data mining and Online Analytical Processing (OLAP) tools are major components of EIP [4]. With the development of decision support activities on Intranet, an easy access to the decision support databases via Intranet is essential to the success of such development. Many management reports and decision support analyses are performed periodically and require the same type of information. Database snapshots are an effective way to meet the needs for such recurring information.

The management of database snapshots for Intranet decision support applications is quite different from that in a traditional transaction processing system. In a transaction processing system, snapshots are typically materialized at a centralized location and are materialized as a regular database file. In an Intranet decision support environment, snapshots may be distributed on users' computers, and may be materialized as a web page rather than a database file. In order to support the OLAP and other decision support applications, snapshots must be converted to a format readable to those applications. The traditional snapshot management commands, such as the DEFINE and the REFRESH commands, only focus on managing the contents of the snapshot. Those commands must be extended to include options for users to specify the snapshot's location

and the format of materialization in order to effectively manage snapshots in this environment.

This paper discusses the special characteristics of the database snapshot management in an Intranet decision support environment, review the major components involved in the management of snapshots, and then define new commands for the management of snapshots in this environment.

2. SNAPSHOT MANAGEMENT IN AN INTRANET DECISION SUPPORT ENVIRONMENT

This section discusses issues related to the management of snapshots in an Intranet decision support environment that make it different from a traditional transaction processing system.

Static Database The database of a transaction processing system is volatile with many updates. A major snapshot management task in that environment is to keep track of updates relevant to snapshots. Unlike transaction databases, a DSDB is updated by a periodical maintenance schedule, and is static between two updates.

Virtual and Materialized Snapshots Snapshots defined on a transaction database must be materialized to maintain the consistency at a point in time. Since a DSDB is static, besides the scheduled maintenance, a DSDB is consistent with its source databases at the time of the last maintenance. Applying the snapshot definition to the DSDB at any time between two scheduled updates will produce results that are consistent with the DSDB and its source databases at the time of the earlier update. Therefore, snapshots that require the same consistent point as the DSDB do not have to be materialized to maintain their consistency. Note that those snapshots may be materialized for efficiency purpose if they are accessed frequently. Hence it is possible to have *virtual* snapshots defined on a DSDB. Snapshots that require a different consistent point from the DSDB must still be materialized.

Support of OLAP and Local Decision Support Applications OLAP is an integral part of EIP [4]. In addition to OLAP applications, users may have other decision support applications installed on their computers. Snapshots with aggregated information are ideal data source for OLAP and other decision support applications. To efficiently support these applications, snapshots may be materialized in a format that is readable by these applications.

Delivery and Materialization Options In an Intranet decision support environment, a user may request a snapshot be delivered (1) as a web page to be viewed with a browser, (2) as a database file downloaded to a users' local database system for further processing, or (3) as a data file readable to a user's decision support applications. Hence a snapshot may also be materialized in those formats. Table 1 illustrates the possible combinations of snapshot materialization options and delivery options. A snapshot materialized as a database file may need to be converted if the user uses a different DBMS. Creating a web page typically requires a server-site scripting such as Microsoft's Active Server Page. Creating other file formats usually requires a specialized program to convert the snapshot to the target format. A virtual snapshot needs querying the database, and then converts the results to the specified format.

Snapshot Ownership and Personalization Some snapshots on Intranet are system-owned and made available to the public; others are user-owned and only for personal use. Web sites today typically provide personalized web pages to users. By keeping track of the snapshot ownership allows the Intranet to provide personalized web pages with links to the user-owned snapshots and the system-owned snapshots that users are allowed to access.

Snapshot Location With an Intranet, a snapshot can be easily downloaded to a user's computer. A user-owned snapshot can be materialized only on a user's computer and not on the server. Based on the location, we can classify snapshots into server-site snapshots and client-site snapshots. Users must decide to create a server-site snapshot or a client-site snapshot. A server-site snapshot is easier to maintain than a client-site snapshot but requires communication costs to access it. A client-site snapshot is locally available but requires more works to keep it up-to-date. The snapshot management in an Intranet environment must include both server-site and client-site snapshots.

Snapshot Refresh Options A snapshot is usually refreshed in response to a user's refresh request, especially for snapshots materialized on their computer. Such a refresh is called *user-pull refresh*. Today, with Internet technology, it is possible for a server to send out refresh messages to clients proactively. Such a refresh is called *server-push refresh*. In a server-push refresh, the server-site snapshot manager refreshes a snapshot automatically. A server-push refresh can be performed every time the DSDB is updated. For snapshots that require constant consistency with the DSDB, the server-push refresh is an ideal way of refreshing.

3. COMPONENTS OF SNAPSHOT MANAGEMENT IN AN INTRANET DECISION SUPPORT ENVIRONMENT

Managing snapshots in this environment requires the coordination between server-site and client-site components. Figure 1 shows the major components and the direction of data flow. The components are introduced in the following.

3.1 Server-Site Components

DSDB Maintenance Algorithm The functions of DSDB maintenance algorithm are well known. Data from the internal and external databases is extracted, cleaned, validated, properly aggregated, and loaded to the DSDB [3]. Changes in the source databases are collected and applied to the DSDB to keep it up-to-date. Since snapshots are defined on the DSDB, the changes prepared by this algorithm can also be applied to update the snapshots.

Server-Site Snapshot Manager The major tasks of snapshot manager are handling requests for creating, refreshing, accessing, and deleting snapshots. It keeps track of a snapshot's definition, consistent point, and the user that created the snapshot. Remembering a snapshot's definition and consistent point enables the snapshot manager to generate all the updates relevant to the snapshot since the consistent point. Remembering a snapshot's owner allows the server to provide security to prevent illegal access to the snapshot, and allows the EIP to provide personalized service such as creating a web page with links to the user's snapshots. Since a snapshot may be delivered to users in various formats, this component must provide appropriate utilities to generate snapshots according to the users' request. These utilities may include server-site scripting tools to generate database web pages, and specialized programs to convert the snapshots from a database format to a decision support application format.

Web Server and EIP This component provides interface for users to access snapshots and perform snapshot management activities, i.e., defining snapshots, refreshing snapshots, etc. A web-enabled query builder with an interface similar to Query-By-Example provides a user-friendly interface for users. For power users who are familiar with database query language, this interface can be a text box to enter the query language command. Ideally, the interface should allow a user to define any type of queries supported by the database server.

3.2 Client-Site Components

Browser A browser provides an interface to the EIP. It enables a user to send queries and receive results. A browser can also be a useful tool for managing snapshots. It has the built-in capability to refresh or reload a dynamic database page, and to save the database page as a local page. Other snapshot management functions may be added to the browser as plug-ins. For example, a browser may be expanded with a *Save AS* plug-in that is able to save the data in a web page to a desirable format.

Personal DSDB Maintenance Algorithm A personal DSDB is owned and used exclusively by the user. It may contain internal and external data as the organizational DSDB. It is personalized to fit a user's need. It may also contain personal knowledge about the problem the user is solving. The internal data may be downloaded from the Intranet or other organizational sources. This component is responsible for maintaining the personal DSDB. It takes data from the Intranet and other sources.

Client-Site Snapshot Manager This component manages snapshots that are downloaded to the client computer. It keeps track of a snapshot's definition and its consistent point in order to refresh the snapshot. In a typical refresh, the client-site snapshot manager initiates the refresh by sending a refresh request to the server-site snapshot manager, and the server-site snapshot manager responses by sending refresh messages to the client-site. The client-site snapshot manager then applies those refresh messages to update the snapshots. The server-site snapshot manager may also initiate the refresh in which it actively sends the refresh messages to the client-site.

4. THE COMMANDS FOR THE MANAGEMENT OF SNAPSHOTS IN AN INTRANET DECISION SUPPORT ENVIRONMENT

From a user's perspective, managing snapshots involves defining, accessing, refreshing and deleting snapshots. This section will define commands to perform these activities in an Intranet decision support environment. It assumes the existence of the server-site and client-site snapshot managers to process these commands, the existence of the EIP on Intranet to provide the interface for users to enter these commands.

Defining snapshots: A snapshot is the realization of the function S{SC(Q, T, D(T)), SM(O, L, M, R)} where SC(A, C, T, D(T)) is a set of parameters specifying the contents of the snapshot, and SM(O, L, M, R) is a set of parameters specifying the management of the snapshot. The meaning of these parameters is explained below:

SC(Snapshot Contents)
- Q: The sequence of query operations and logical conditions to generate snapshot records
- T: A point in time called the consistent point
- D(T): The database state at T

SM(Snapshot Management)
- O: Snapshot owner
- L: Snapshot location
- M: Snapshot materialization option
- R: Snapshot refresh option

The following DEFINE SNAPSHOT command lets users to enter parameters:

DEFINE SNAPSHOT *<snapshot-name>*
AS *<query>*
[AS OF *consistent-point*]
[OWNER IS *system* | *username*]
[{MATERIALIZED AT *server-site* | *client-site*
AS *database file* | *web page* | *other files* }]

[REFRESH BY user-*push* | server-*pull*]

In the syntax, clauses enclosed in the brackets are optional; clauses enclosed in the braces may be repeated; words in italic separated by the vertical bars are possible choices for the parameters.

The AS OF clause specifies that the snapshot represent the state of the database at the time of the consistent point. This kind of consistency is called point consistency. There are two types of consistent point: (a) System-determined consistent point: This is the case where the DSDB determines the consistent point. Since the DSDB is basically a snapshot of the source databases, the DSDB is consistent with the source databases at the time the DSDB was last updated in a scheduled maintenance. Hence, when a snapshot is created, its consistency point is the time the DSDB was last updated. (b) User-specified consistent point: This is the case where users determine the consistent point. Frequently users may need a snapshot with a consistent point different from the DSDB database to fit their need. For example, rather than using an end-of-day consistent point determined by the system, a user may request a snapshot that is consistent with the source databases at the end of half-of-day. Without the AS OF clause, the system-determined consistent point is assumed.

The OWNER IS clause specifies whether the snapshot is either system or user-owned. Without it, the system-owned is assumed.

The MATERIALIZED AT clause specifies the snapshot's location and its format of materialization. Without it, the server-site is assumed. Note that the *other files* option is a list of other file formats supported by the system. This clause can be repeated which indicates a snapshot can be materialized both at the server-site and at the client-site, optionally with multiple formats. Hence, it is possible to enter the following clauses in a DEFINE SNAPSHOT command:

MATERIALIZED AT server-site AS database file
MATERIALIZED AT client-site AS database file
MATERIALIZED AT client-site AS web page

The REFRESH BY clause specifies the snapshot to be refreshed by a server-push or user-pull refresh. Without it, the user-pull refresh is assumed.

Accessing Snapshots As discussed earlier, a user may request a snapshot to be delivered as a database file, a web page, or as a file readable to the decision support applications. This may require a specialized program to do the conversion, and needs the support of a web server. Therefore, a sever-site snapshot can be retrieved with a format different from its original materialized format. A client-site snapshot without the support of a web server can only be accessed in its original format, and typically via a link on a web page. Hence, the RETRIEVE command proposed below applies only to the server-site snapshots.

RETRIEVE <*snapshot-name*>
[{AS *database file* | *web page* | *other files* }]

The AS clause lets users specify the retrieved format which may be different from the original format, and can be repeated. Hence, it is possible to enter the following clauses in a RETRIEVE command:

AS database file
AS web page

Without it, the web page is assumed.

Refreshing Snapshots As discussed in the previous section, a snapshot specifies with a server-push refresh in its definition does not require users to send refresh request. The following REFRESH command applies only to the user-pull refresh.

REFRESH <*snapshot-name*>
[AS OF *consistent-point*]

The AS OF clause allows the user to specify the new the consistent point, of which may be different from the system-determined point. Without it, the system-determined consistent point is assumed.

Deleting Snapshots: Deleting a server-site snapshot requires removing its definition from the server-site manager. Deleting a client-site snapshot requires removing the definition from both the server-site and client-site snapshot manager. The delete command is simply:

DELETE <*snapshot-name*>

5. CONCLUSIONS

Database snapshots have been an effective way of supporting decision support applications. Today, Intranet with EIP have become an important platform to provide decision support services and data communication. Hence, snapshots will continue to play an important role in providing data in an Intranet decision support environment. This paper addresses the management of database snapshots in such environments from users' perspectives. Managing snapshots involves defining, accessing, refreshing and deleting snapshots. I have defined commands to perform these activities. Although these commands have not been implemented, they illustrate the ideal capabilities that such commands should offer to the users.

REFERENCES

[1] Adiba, M. & Lindsay, B. (1980). Database snapshots. Proceedings of the 6th International Conference on Very Large Data Bases, pp. 86-91.

[2] Labrinidis, A. & Roussopoulos, N. (2000). Webview Materialization. ACM SIGMOD International Conference on Management of Data, May 14-19, 2000.

[3] Gray, P., Watson, H. (1998). Decision Support in the Data Warehouse. Prentice Hall, 1998.

[4] Murray, G. (1999). Making Connections with Enterprise Knowledge Portals. White Paper, Computerworld, Sept. 6, 1999.

TABLE 1: POSSIBLE COMBINATIONS OF SNAPSHOT MATERIALIZATION AND DELIVERY OPTIONS

		Materialization Options			
		Database File	**WebPage**	**Other File Formats**	**Virtual Snapshot**
Delivery Options	**Database File**	Ready/ Conversion	Conversion	Conversion	Querying & Conversion
	Web Page	Server site scripting	Ready	Server site scripting	Querying & Server site scripting
	Other Formats	Conversion	Conversion	Ready / Conversion	Querying & Conversion

FIGURE 1: COMPONENTS OF SNAPSHOT MANAGEMENT IN AN INTRANET DECISION SUPPORT ENVIRONMENT.