τ-Synopses: A system for run-time management of remote synopses

Yossi Matias
School of Computer Science
Tel Aviv University
matias@cs.tau.ac.il

Leon Portman
School of Computer Science
Tel Aviv University
leonpo@cs.tau.ac.il

Abstract

τ-Synopses is a system designed to provide a run-time environment for remote execution of various synopses. It enables easy registration of new synopses from remote platforms, after which the system can manage these synopses, including triggering their construction, rebuild and update, and invoking them for approximate query processing. The system captures and analyzes query workloads, enabling its registered synopses to significantly boost their effectiveness (efficiency, accuracy, confidence), by exploiting workload information for synopses construction and update. The system can also serve as a research platform for experimental evaluation and comparison of different synopses.

1 Motivation

Approximate query processing provides important alternatives to existing relational databases when exact query answers are not required. This task is usually done by using *data synopses*, consice representations of data sets, such as histograms, splines, sampling, wavelets or other methods.

Increased interest in approximate query processing resulted with proliferation of new synopses addressing new problems as well as proposed alternatives to previously suggested synopses. In particular, synopses are becoming more advanced, supporting updates to data, awareness to workload, and adaptive to changes in workload (e.g., [1, 4, 5, 6, 7, 8, 9, 10, 11, 15]).

Several systems and projects address approximate query processing and data synopses. In the AQUA Project [12, 3, 2], synopses are precomputed and stored in a DBMS. It provides approximate answers by rewriting the queries to run on these synopses, and enables keeping synopses up-to-date as the database changes. Various aspects of approximate query processing were studied by Microsoft Research DB group (e.g., [6, 8]). The question of how to reconcile various synopses for large information sources with many tables was studied in [14, 13].

Operational systems requires the management and consolidation of many synopses. These include various synopses addressing different types of queries, each requiring its own type of synopsis; furthermore, even for the same type of query, many different instances of the same synopsis should be used to represent different data sets (e.g., different relations, or different columns of the same relation). Finally, in some cases it would be beneficial to hold more than one synopsis for the same type of queries, to benefit from the different properties of the various synopses (e.g., having an answer based on all supporting synopses, letting each synopsis support a particular subset of the queries etc).

For research purposes, it would also be beneficial to support many synopses. Indeed, for each query there may be different possible types of synopses that could be useful. Furthermore, for each particular synopsis, it would be useful to compare different implementations for evaluation and benchmarking purposes. Therefore, it is advantageous to have a system that can accommodate *multiple synopses*, and have an easy way to integrate new synopses and manage them.

The multiple synopses in use in either operational or research system could be placed in remote locations for various reasons: they may be implemented on different types of platforms, they may be summarizing remote data whose transfer is undesirable or impossible due to performance or security constraints, and it would be beneficial to share the load of operating a large number of synopses using different systems for load balancing and redundancy reasons. Therefore, it would be beneficial to have a system support *remote execution* of registered synopses.

Synopses are more effective when they are adaptive for predicted workload, changes in workload, changes in data, and changes in performance requirements (query time, accuracy, confidence, and available memory). This motivates the use of a system of *managed synopses*. This is a single system that registers all synopses, triggers their construction and maintenance (in response to new data, data changes, predicted changed workload, or by demand). Having a managed synopses system enables providing information required by all relevant synopses from a single repository (e.g., data, workload, changes in data and workload).

2 τ-Synopses framework overview

Motivated by the above, the τ-Synopses system was designed to provide a run-time environment for remote execution of various synopses. It enables easy registration of new synopses from remote SOAP-enabled platforms, after which the system can manage these synopses, including triggering their construction, rebuild and update, and invoking them for approximate query processing. The system captures and analyzes query workloads, enabling its registered synopses to significantly boost their effectiveness (efficiency, accuracy, confidence), by exploiting workload information for synopses construction and update. The system can serve as a research platform for experimental evaluation and comparison of different synopses.

The τ-Synopses is a stand alone system, and can work with data sources such as existing relational or other database systems. It supports two types of users: synopses providers who register their synopses within the system, and end-users who send queries to the system. The system administrator defines available data sources and provides general administration of the system.

When a new synopsis is registered, the relevant data set and the supported queries are defined. A query submitted to the system is executed using the appropriate synopsis, based on the registration and other information. The result is returned to the user or optionally processed by other modules in the system. The system transforms updated data from its original datasource to be consistent with the format known to the synopses, so that synopses are not required to support any data transformation functionality or database connectivity logic. Any relational database or even real-time data providers can be datasources in the system.

Workload information is recorded by the system and becomes available to the registered workload-sensitive synopses.

The τ-Synopses system has the following key features:

- *multiple synopses*: The system can accommodate various types of synopses. New synopses can be added with their defined functionalities.

- *pluggable integration*: For integration purposes, a synopsis has to implement a simple interface, regardless of its internal implementation. By utilizing a light-weight host provided by the system, the synopsis can be executed on any SOAP-enabled platform.

- *remote execution*: Synopses can be transparently executed on remote machines, over TCP/IP or HTTP protocols, within local area networks or over the internet.

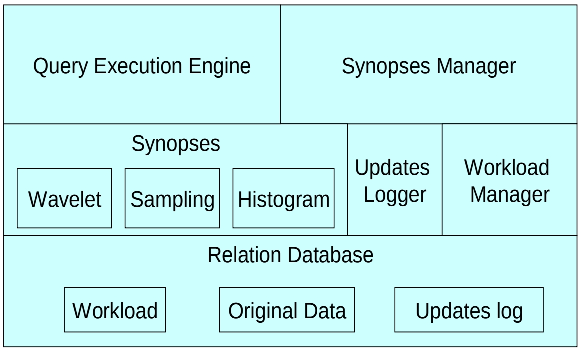


Figure 1. Synopses Framework Architecture

- *managed synopses*: The system allocates resources to synopses, triggers their construction and maintenance, selects appropriate synopses for execution, and provides all required data to the various synopses.

- *workload support*: Workload is captured, maintained and analyzed in a centralized location, and made available to the various synopses for construction and maintenance.

- *research platform*: The system provides a single, consistent source of data, training and test workload for experimental comparison and benchmarking, as well as performance measurements. It can therefore serve as an effective research platform for comparing different synopses without re-implementing them.

3 Architecture

The core of the τ-Synopses system architecture features the following components, and depicted in Figure 1: Query Execution Engine, Synopses Manager, Updates Logger, and Workload Manager. In addition, it includes a query-application which is used by end-users, an administration-application used by the administrator and by synopses-providers, and a pool of registered synopses.

The Synopses Manager is used for registration and maintenance of the synopses. A new synopsis is added to the system by registering its parameters (including list of supported queries and data sets) in the Synopses Manager Catalog.

The Query Execution Engine provides interface for receiving query request from end-users and invoking the appropriate synopsis (or synopses), as determined by the Synopses Manager in order to process such query.

The Updates Logger provides all data updates to the registered synopses by intercepting data updates information in the data sources.

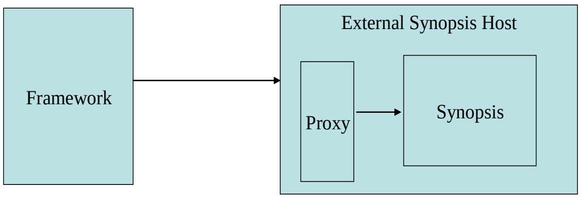


Figure 2. Synopses Integration

The Workload Manager captures, maintains and analyzes workload information for building, maintaining and testing synopses.

Figure 2 depicts integration process of a remote synopsis within the framework. The system provides a light-weight host process, inside which the custom synopsis will be running. The host is responsible for all communication with the system and is transparent to the synopsis. This design enables unconstraint deployment. A remote synopsis can be integrated into the system by deploying or adapting such host into the remote system, and connecting the synopsis module locally into the host.

Figure 3 illustrates an overall view of the system in a distributed environment, consisting of multiple remote synopses, each representing its local data source.

4 System implementation

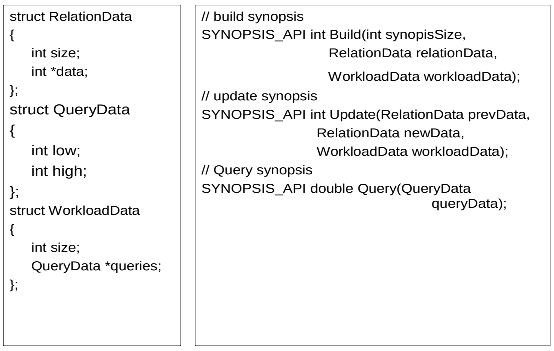
The system modules were implemented in the .NET framework, with remote modules communicating through the .NET Remoting. In order to integrate a new synopsis, it is sufficient to have it implemented on a SOAP-enabled platform. Figure 4 shows the interfaces required for the basic functionality. After incorporating these interfaces into a synopsis, it can already be used in the system.

The system was tested by having groups of students implement remote synopses as part of their projects, and have these synopses connect to the core system using the simple interfaces. We will encourage other research groups connect their synopses to to the τ-Synopses system.

5 Demo Scenario

The demo will showcase the system's components: its core, the query-application, the administrator-application, and a number of simple remote synopses.

For the demo, we will use METAR data reports, provided by National Weather Service. These reports include surface weather observations that are the basic information upon which forecasts and warnings are made in support of a wide range of weather sensitive activities within public and private sectors. METAR contains a report of wind, visibility, runway visual range, present weather, sky condition, temperature, dew point, and altimeter settings. These observations are published every hour in the wether.noaa.gov web site, and cover a total of 7500 reported locations. The data is maintained in a circular fashion, so that at any time, only the data for the last 24 hours is available.



```
struct RelationData
{
    int size;
    int *data;
};
struct QueryData
{
    int low;
    int high;
};
struct WorkloadData
{
    int size;
    QueryData *queries;
};
```

```
// build synopsis
SYNOPSIS_API int Build(int synopisSize,
                       RelationData relationData,
                       WorkloadData workloadData);
// update synopsis
SYNOPSIS_API int Update(RelationData prevData,
                     RelationData newData,
                     WorkloadData workloadData);
// Query synopsis
SYNOPSIS_API double Query(QueryData
                                  queryData);
```

Figure 4. Basic required interfaces

This data has a number of interesting characteristics for the demo: dynamic updates, good correlation of data values, non-uniform query workloads, and data that becomes rapidly unavailable.

We will demonstrate: (i) how to define a datasource for METAR data; (ii) how to register and run remote synopses; (iii) how to execute benchmarks on different synopses; (iii) how users can execute approximate queries on synopses (of unavailable data source); and (iv) how synopses adapt to the changes in the underlying data and query workload.

References

[1] A. Aboulnaga and S. Chaudhuri. Self-tuning histograms: building histograms without looking at data. In *Proceedings of the 1999 ACM SIGMOD*, pages 181–192, 1999.

[2] S. Acharya, P. Gibbons, and V. Poosala. Aqua: A fast decision support system using approximate query answers. In *Proceedings of the 1999 VLDB*, 1999.

[3] S. Acharya, P. Gibbons, V. Poosala, and S. Ramaswamy. The aqua approximate query answering system. In *Proceedings of the 1999 ACM SIGMOD*, 1999.

[4] S. Acharya, P. B. Gibbons, and V. Poosala. Congressional samples for approximate answering of group-by queries. In *SIGMOD Conference*, pages 487–498, 2000.

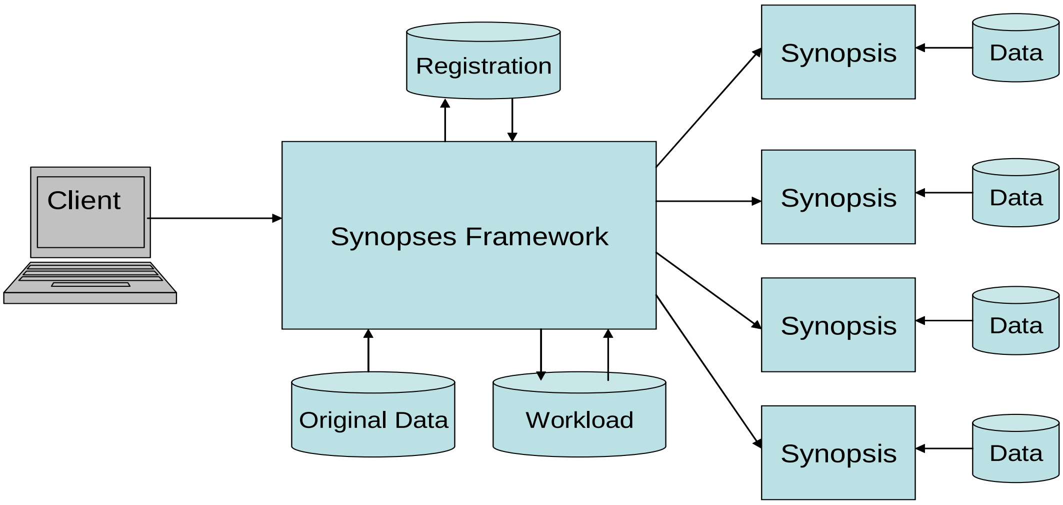


Figure 3. Distributed Environment

[5] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. Join synopses for approximate query answering. In *Proceedings of the 2001 ACM SIGMOD*, pages 275–286, 1999.

[6] N. Bruno, S. Chaudhuri, and L. Gravano. STHoles: a multidimensional workload-aware histogram. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 30(2):211–222, 2001.

[7] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim. Approximate query processing using wavelets. *VLDB Journal: Very Large Data Bases*, 10(2–3):199–223, 2001.

[8] S. Chaudhuri, G. Das, and V. Narasayya. A robust, optimization-based approach for approximate answering of aggregate queries. In *Proceedings of the 2001 ACM SIGMOD*, 2001.

[9] V. Ganti, M.-L. Lee, and R. Ramakrishnan. ICICLES: Self-tuning samples for approximate query answering. In *The VLDB Journal*, pages 176–187, 2000.

[10] M. Garofalakis and P. B. Gibbons. Wavelet synopses with error guarantees. In *Proceedings of the 2002 ACM SIGMOD*, pages 476–487, 2002.

[11] P. B. Gibbons and Y. Matias. New sampling-based summary statistics for improving approximate query answers. In *Proceedings of the 1998 ACM SIGMOD*, pages 331–342, 1998.

[12] P. B. Gibbons, Y. Matias, and V. Poosala. AQUA project white paper. Technical report, Bell Labs, Murray Hill, New Jersey, U.S.A., 1997.

[13] A. C. Konig and G. Weikum. Automatic tuning of data synopses. *Information Systems*, 28(1-2). Special Issue of papers from EDBT 2002.

[14] A. C. Konig and G. Weikum. A framework for the physical design problem for data synopses. March 2002.

[15] Y. Matias, J. S. Vitter, and M. Wang. Dynamic maintenance of wavelet-based histograms. In *The VLDB Journal*, pages 101–110, 2000.