Tutorial

Using the AMBER Data Repository to Analyze, Share and Cross-exploit Dependability Data

Marco Vieira
mvieira@dei.uc.pt
University of Coimbra, Portugal
The Second International Conference on Dependability (DEPEND 2009)
Athens/Glyfada, Greece, June 18,2009

The AMBER Project

- Assessing, Measuring and Benchmarking Resilience in computer systems and components (AMBER)
- Coordination Action supported by the European Commission in the 7th FP
- Coordinating and advancing research in resilience measurement and benchmarking in computer systems and infrastructures

Current challenges

- Quality of measurements
- Integration of the human and technical components of the analysis
- Dynamic and adaptive systems and networks
- Integration with the development processes

AMBER objectives

- State-of-the art survey
- Research agenda
- Data repository
- Others:
  - Dissemination events (workshops, panels, etc)
  - Benchmarking tools
  - Training material

This Tutorial…

Learn how to use the AMBER Data Repository to analyze and share data from dependability evaluation experiments

Problems

- How to **analyze** the usually large amount of raw data produced in dependability evaluation experiments?
- How to **compare** results from different experiments or results of similar experiments across different systems?
  - Different and incompatible tools, data formats, and setup details…
- How to **share** raw experimental results among research teams?

Current situation

- The situation today is not good!!!
- Spreadsheets and other specific tools to analyze results
  - Not standard and difficult to build
- Difficult to compare data and generalize conclusions
- Researchers share final results and conclusions
  - Papers, mainly
  - Raw data is not shared

ADR Vision and objectives

- Vision
  - Become a worldwide repository for dependability related data
- Key objectives:
  - Provide state-of-the-art data analysis
  - Allow data comparison and cross-exploitation
  - Facilitate worldwide data sharing and dissemination
- Potential tool to increase the impact of research

Data analysis approach

- Repository to analyze, compare, and share results
- Use a business intelligence approach:
  - Data warehouse to store data
  - On-Line Analytical Processing (OLAP) to analyze data
  - Data mining algorithms to identify (unknown) phenomena in the data
  - Information retrieval for data in textual formats
- Adopt the same life cycle of BI data

Outline

1. Business Intelligence
2. Data Warehousing & OLAP
3. Using DW to analyze dependability related data
4. The AMBER Data Repository

1. Business Intelligence

What is Business Intelligence?

- Business Intelligence (BI):
  - **Getting the right information, to the right decision makers, at the right time**
- BI is an enterprise-wide platform that supports, data gathering, reporting, analysis and decision making
- BI is meant to:
  - Fact-based decision making
  - "Single version of the truth"
- BI includes reporting and analytics

Five classic BI questions

- What happened?
- What is happening?
- Why did it happen?
- What will happen?
- What do I want to happen?

Past

Present

Future

Typical BI technologies

- ETL Tools (Extract, Transform, and Load)
- Repositories
  - Data Warehouse
- Analytical tools
  - Reporting and querying
  - OLAP
  - Data mining
- Information retrieval

Many proprietary products

ACE*COMM
Ab Initio
Actuate
ComArch
CyberQuery
Dimensional Insight
IBM
  Applix
  Cognos
InetSoft
Informatica
Information Builders
LogiXML
LucidEra
MicroStrategy
Microsoft
  Microsoft Analysis Services
  PerformancePoint Server 2007
  Proclarity
Oracle Corporation
  Hyperion Solutions Corporation
Panorama Software
Pentaho
Pervasive
Pilot Software, Inc.
PRELYTIS
Prospero Business Suite
Qliktech
SAP Business Inf. Warehouse
Business Objects
  OutlookSoft
SAS Institute
Siebel Systems
Spotfire (now Tibco)
StatSoft
SPSS
Telerik Reporting
Teradata
Thomson Data Analyzer

Some open source/free producs

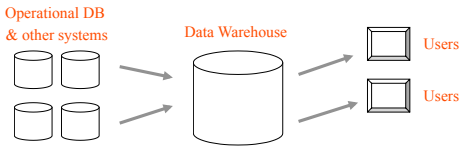
- Eclipse BIRT Project:.
- Freereporting.com:
- JasperSoft:
- OpenI:
- Palo (OLAP database):
- Pentaho:
- RapidMiner
- SpagoBI:
- Weka

- Some products from big companies can be used freely

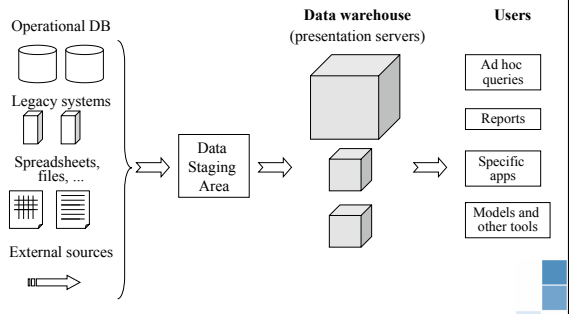
2. Data Warehousing & OLAP

What is a Data Warehouse?

- Big database that stores data for decision support
- Built from the operational data collected from transactional DB and other operational systems



Basic DW components



Data volume

- Less than 20 GBytes
  - Small dimension; runs in a PC
- From 20 to 100 GBytes
  - Medium dimension; needs a powerful workstation
- From 100 Gbytes to 1 TBytes
  - Large dimension; needs a powerful server, normally with parallel processing
- More than 1 TBytes
  - Very large dimension; massive parallel processing

Some characteristics

- Temporal dependency
- Non volatile
- Target oriented
- Data integration and consistency
- Designed for queries

Temporal dependency

- The data is collected over time
  - Do not represent a specific moment
  - Represents the history
- A temporal reference must be associated to all data in the database

Non volatile

- The data in the DW is never updated
- The DW stores historic data (historic memory) collected from the operational databases
- After being load (from the operational databases) there is only one operation:
  - Queries

Target oriented

- The data warehouse must only store data relevant for decision support
- Many operational data (needed for everyday management) is not relevant for the DW

Data integration and consistency

- In a operational environment the information may be stored in different locations using different representations
- That data must be integrated and made consistent before being load in the DW

Designed for queries

- After being load the data never changes:
  - Only queries are allowed
- DW stores a large amount of data

The data must be stored in such a way that improves performance

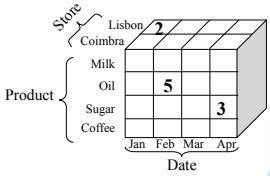
Multidimensional view
Partial denormalization

Dimensional model

- Typical model in operational databases: E/R
- The dimensional model follows a different approach
  - Stores the same data
  - Data organization is user oriented
    - Easy to understand
    - Very good performance for queries
- Data Warehouses built over complex E/R models never succeed

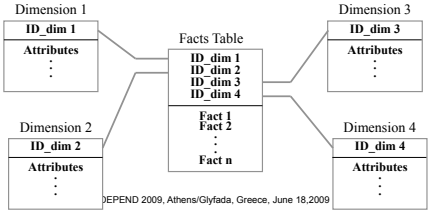
The multidimensional model

- Facts stored in a multidimensional array
- The dimensions are used to index the array
- Usually built using data from operational databases



Star model

- The typical dimensional model is a star structure with:
  - A central table with facts
  - Several dimensions tables describing the facts



Facts

- Represent the business measures
- The most useful facts are:
  - Numbers
  - Additives

Facts table

- Comprises several numeric attributes (facts) and foreign keys to the dimensions
- Normalized table
- Relationships M:1 with the business dimensions
- Contains normally a large number of records
- Represents typically 95% of the space used by the DW

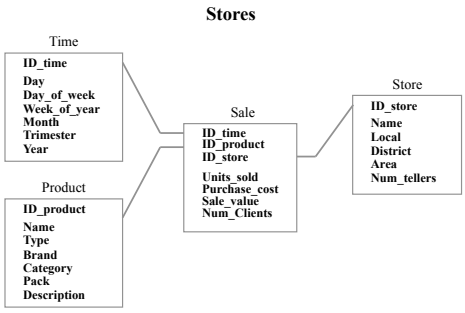
Dimensions

- Each dimension represents a business parameter
  - Time, clients, products, etc
- Represent a entry point for the analysis of the facts
- Represent different point-of-views for the analysis of the facts

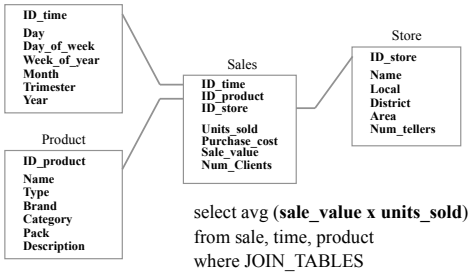
Dimension tables

- Strongly denormalized
  - For performance
- Dimensions have hierarchies
  - Day ➔ Month ➔ Year ➔ … Contain a large set of attributes
- Typically comprise a small number of records (when compared to the facts table)

Star schema example



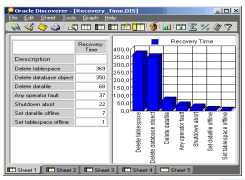
Low level queries



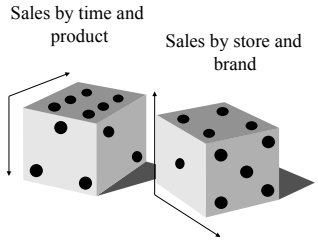

```
select avg (sale_value x units_sold)
from sale, time, product
where JOIN_TABLES
group by brand, month
```


User interfaces

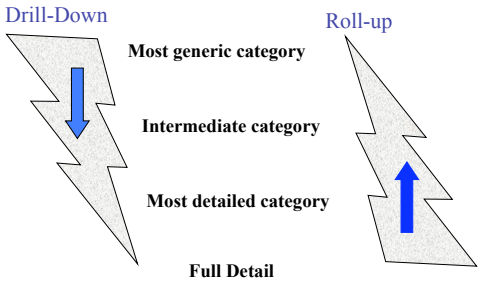
- Explore data in Data Warehouses
  - Typical OLAP tools
    - Access the relational engine using SQL
    - Data presentation using tables, graphics, reports, etc
    - Targeted for ad-hoc queries
  - Other tools
    - Data mining
    - Modeling



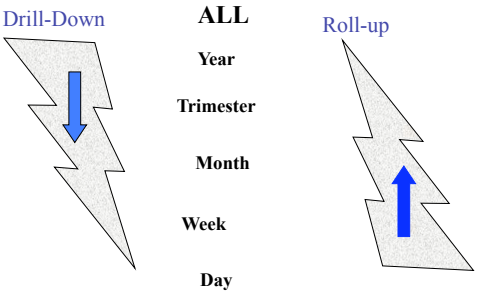
Queries - Slice and Dice



Drill-Down & Roll-Up



Time: Drill-Down & Roll-Up



Steps for the design of the star model

1. Identify the business process/activity
2. Identify the facts
3. Identify the dimensions
4. Define the data granularity
   - Day, Week, Month, …
   - Product, Category, …
   - Store, City, …

Do not forget that the model depends on the data available (operational databases, files, etc)

Example – Retail sales

- Set of stores belonging to the same enterprise
- Goal: Analysis of sales
- Each store has several departments (food, hygiene and cleaning, etc)
- Sells thousands of products
- Products are identified using a unique number

Retail sales – Business data

- Where to collect the data?
  - POS - point of sales
  - Operational database
- What to measure?
  - Sales
- Goals?
  - Maximize the profit
  - Maximum sales price possible
  - Lower costs
  - More clients

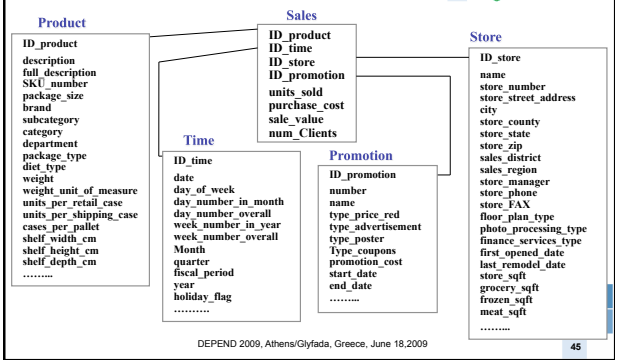
Retail sales – Facts

- Examples of relevant decision support facts:
  - Number of units sold
  - Acquisition costs
  - Sale value
  - Number of clients that bought the product
- Question: is it possible to obtain base data (from the operational system) for these facts?

Retail sales – Dimensions

- Main dimensions:
  - Product x Store x Time
- Are there other relevant dimensions?
  - Supplier? – Promotions? – Client?
  - Employee responsible for the store on that day?
- It is normally possible to add extra dimensions
- All the dimensions have a 1:M relationship with the facts

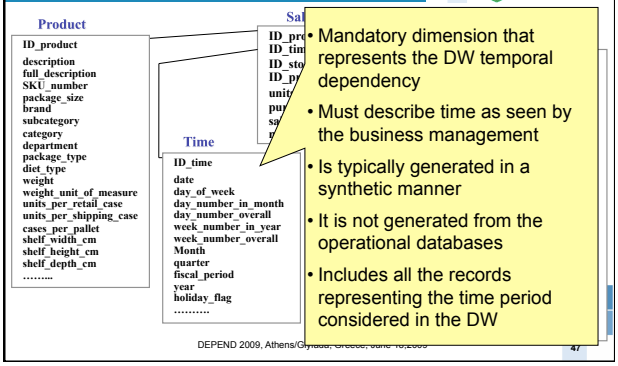
Retail sales



Granularity

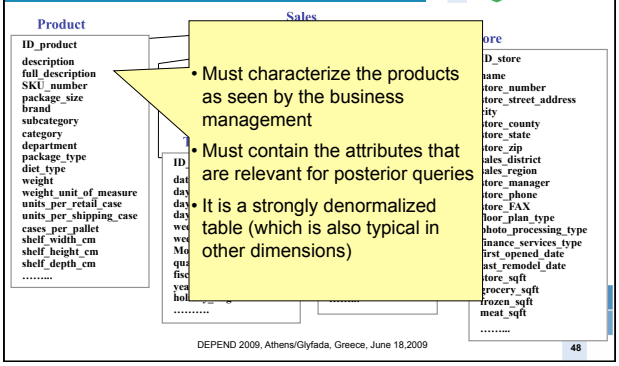
- Example: record the daily sales for all products
  - Analyze in detail (price, quantity, etc) the products sold every day, in each store, …
- Retail sales granularity:
  - Products x Store x Promotion x Day
- The granularity defines the detail of the DW and has a strong impact in the size
- The granularity must be adjusted to the analysis requirements

Retail sales – Details



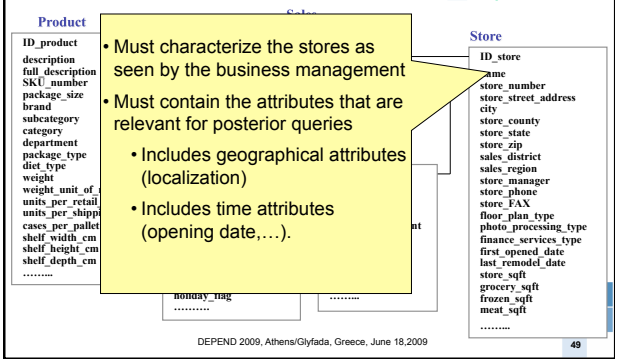
- Mandatory dimension that represents the DW temporal dependency
- Must describe time as seen by the business management
- Is typically generated in a synthetic manner
- It is not generated from the operational databases
- Includes all the records representing the time period considered in the DW

Retail sales – Details



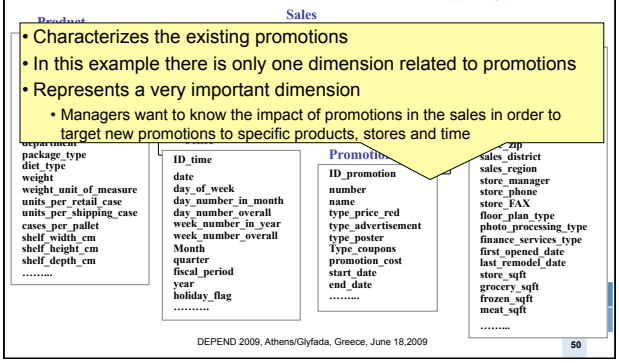
- Must characterize the products as seen by the business management
- Must contain the attributes that are relevant for posterior queries
- It is a strongly denormalized table (which is also typical in other dimensions)

Retail sales – Details



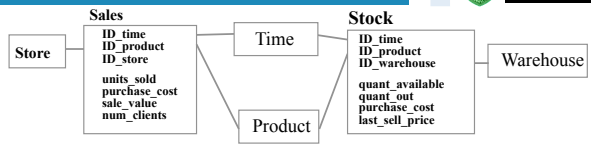
- Must characterize the stores as seen by the business management
- Must contain the attributes that are relevant for posterior queries
  - Includes geographical attributes (localization)
  - Includes time attributes (opening date,…).

Retail sales – Details



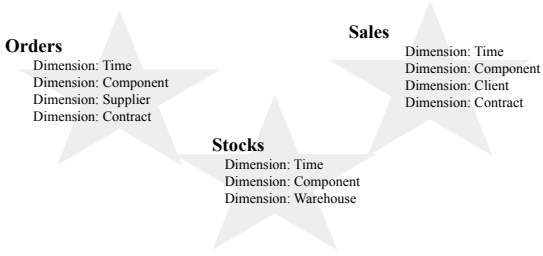
- Characterizes the existing promotions
- In this example there is only one dimension related to promotions
- Represents a very important dimension
  - Managers want to know the impact of promotions in the sales in order to target new promotions to specific products, stores and time

More than one star



- Two or more starts can be connected using one or more dimensions
- Shared dimensions must be conform
  - Contain consistent data when considering each star
- **Drill across**: query that crosses more than one start

Several stars

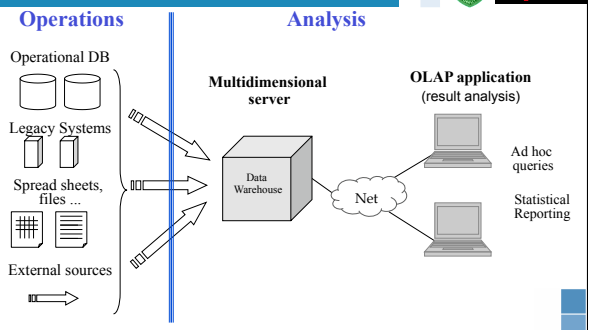


Questions

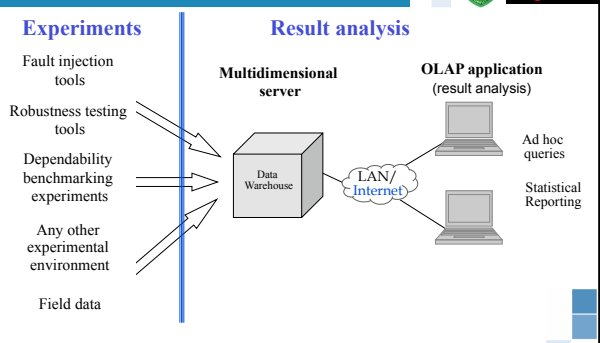
?

3. Using DW to analyze dependability data

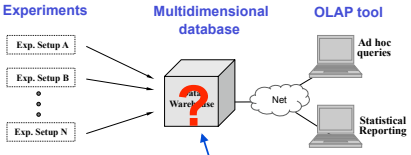
Basic elements of a DW



A DW for experimental data

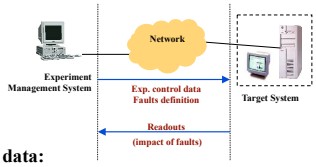


Key points of the proposed approach



- General approach to store results from dependability evaluation experiments
- Data from different experiments can be compared/cross-exploit (only if it makes sense to compare)
- Raw data is available (not only the final results)
- Results can be analyzed and shared world wide by using web-enabled versions of OLAP tools

Two types of data in experimental dependability evaluation

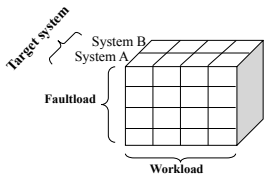


Two types of data:

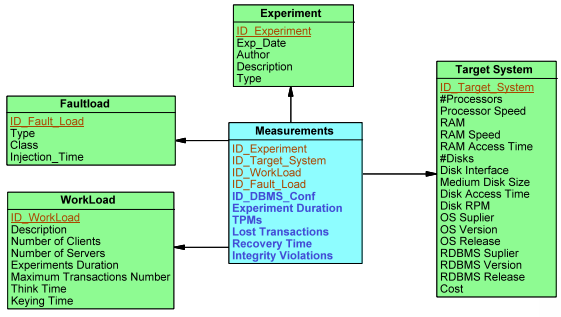
- Measures collected from the target system (**FACTS**)
  - For example, raw data representing error detection efficiency, recovery time, failure modes, etc
- Features of the target system and experimental setup that have impact on the measures (**DIMENSIONS**)
  - For example, attributes describing the target systems, the different configurations, the workload, the faultload, etc

The multidimensional model

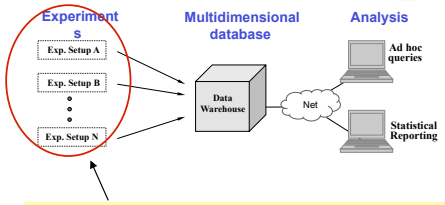
- Facts are stored in a multidimensional array
- Dimensions are used to access the array according to any possible criteria



The star schema



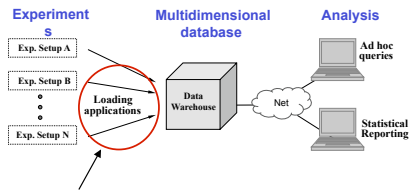
Basic elements of the proposed approach



The experimental setups are used as they are. You can use your favorite dependability evaluation tool and do the experiments in the usual way. It's necessary…

- To know the format of the raw results
- To have access to the results

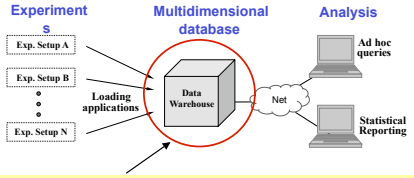
Basic elements of the proposed approach



Loading applications

- General purpose loading applications
- Some transformations in the data are normally necessary for consistency

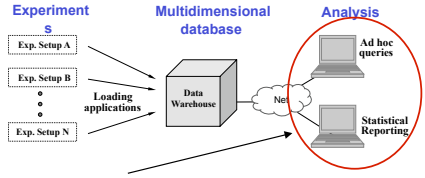
Basic elements of the proposed approach



Data warehouse

- Raw data is available in a standard star schema (facts + dimensions)
- Results from different experiments are compatible and can be compared/ analyzed together, then they are stored in the same star schema (or in scheme that share at least one dimension)
- If results are from different unrelated experiments then they are stored in a separated schema

Basic elements of the proposed approach



Analysis

- Commercial OLAP tools are used to analyze the raw data and compute the measures. These tools are designed to be used by managers: very easy to use :-)
- Just need an internet browser to analyze the data

Steps needed to put our approach into practice

1. Definition of the adequate star schema to store the data. Create the tables in the data warehouse
2. Use general-purpose loading application to define the loading plans for each table in the star schema
3. Run the loading plans to load the star tables with the raw data collected from the experiments
4. Every time a new experiment is done corresponding loading plans are run again to add the new data to the data warehouse
5. Analyze the data: calculate measures, find unexpected results, analyze trends, etc

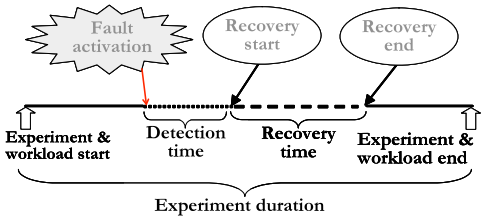
Example: Recovery and Performance Evaluation in DBMS

- Tuning of a large DBMS is very complex
- Administrators tend to focus on performance tuning and disregard the recovery features
- Administrators seldom have feedback on how good a given configuration is
- A technique to characterize the performance and the recoverability in DBMS is needed

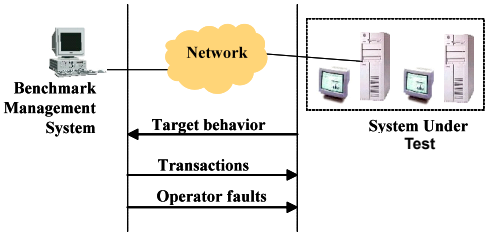
The Approach

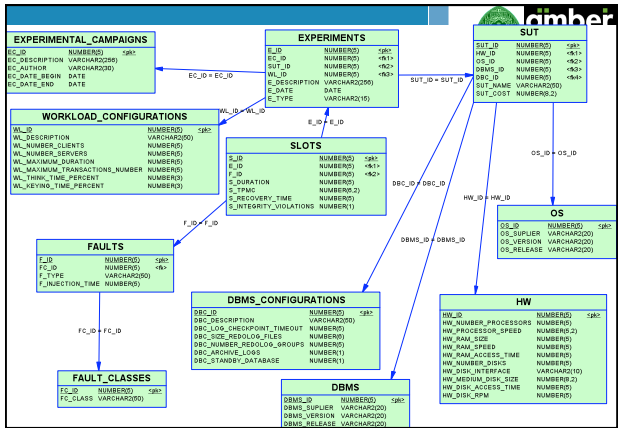
- Extending existing performance benchmarks to evaluate recoverability features in DBMS
- Include a faultload and new measures
- Faultload based on operator faults
- Measures related to recovery:
  - Recovery time
  - Data integrity violations
  - Lost transactions

Operator faults injection and recovery



Experimental setup





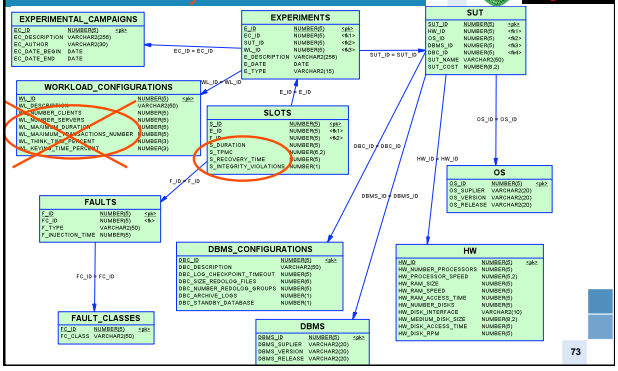
Steps towards data analyzes

1. Definition of the adequate star schema
   a. Identify the process/activity
   b. Identify the facts
   c. Identify the dimensions
   d. Define the data granularity
2. Load the data
3. Analyze the data

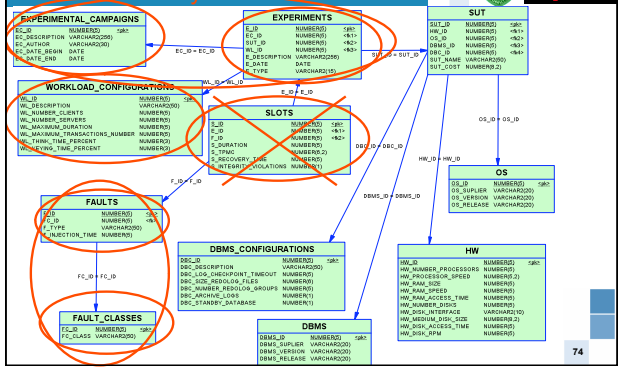
Definition of the adequate star schema: Identify the process/activity

- Experiments to characterize the performance and the recoverability in DBMS
- Includes a faultload and new measures
- Faultload based on operator faults
- Measures related to recovery

Definition of the adequate star schema: Identify the facts



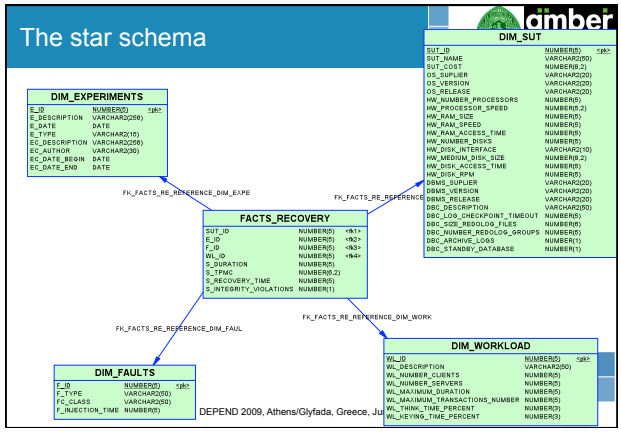
Definition of the adequate star schema: Identify the dimensions



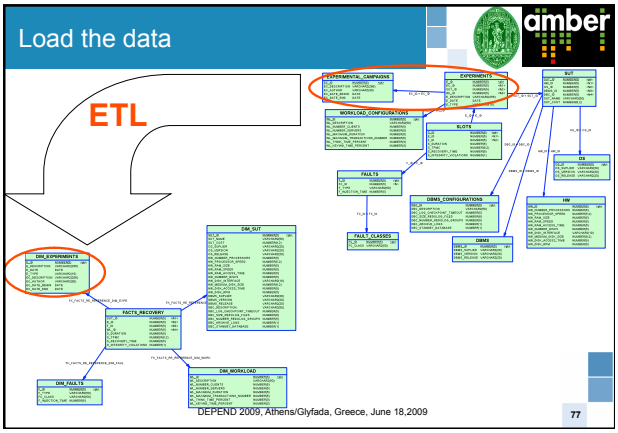
Definition of the adequate star schema: Define the data granularity

- Performance and recovery results
  - Per experiment
  - Per SUT
  - Per workload
  - Per fault type

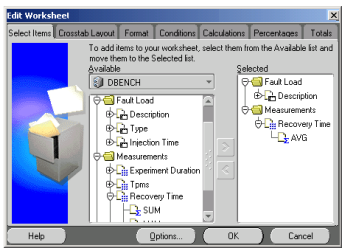
The star schema



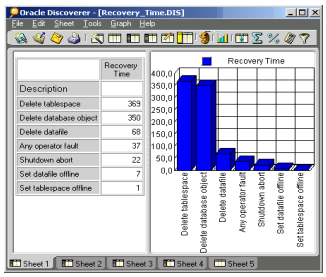
Load the data



Analyze the data: Example of query construction



Analyze the data: Example of query answer



Questions

?

4. The AMBER Data Repository

AMBER Repository vision and objectives

- Vision
  - Become a worldwide repository for dependability related data
- Key objectives:
  - Provide state-of-the-art data analysis
  - Allow data comparison and cross-exploitation
  - Facilitate worldwide data sharing and dissemination
- Potential tool to increase the impact of research

Potential use

- Research team level
  - Perform the analysis of data in an efficient way
  - Efficient dissemination of the results of the team
- Project level
  - Sharing and cross-exploitation of results from different project teams
- World wide
  - Common repository to store and share data
  - Many teams are performing dependability evaluation but there are no results available at the web

Data analysis approach

- Repository to analyze, compare, and share results
- Use a business intelligence approach:
  - Data warehouse to store data
  - On-Line Analytical Processing (OLAP) to analyze data
  - Data mining algorithms to identify (unknown) phenomena in the data
  - Information retrieval to access data in textual formats
- Adopt the same life cycle of BI data
- Use technology already available for DW, DM & IR

Steps

1. User registration
2. Multidimensional analysis
3. Definition of the loading plans
7. Load the data
8. Definition of data ownership policies
9. Analysis of the data

- Analyze DBench-OLTP results using OLAP

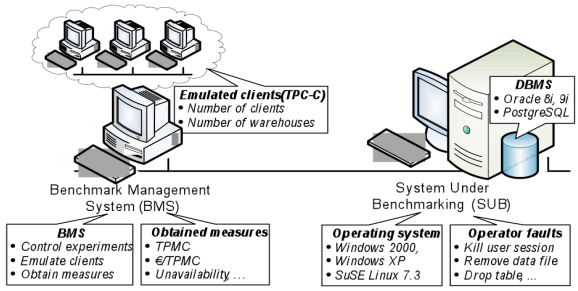
User registration

- ADR users must undergo a registration procedure
- Provide identification information that is verified by the ADR support team
  - To filter malicious users
- Contact information is used to get in touch with the potential repository user
- To access the repository users must authenticate

Multidimensional analysis

- Design an adequate multidimensional data model
- User has the required expertise to design the data model ☺
  - Send to the ADR support team the SQL scripts needed to create the database tables
- The ADR team helps the user defining the model
  - The user only needs to explain us the experimental setup and the format of the data collected

The DBench-OLTP benchmark



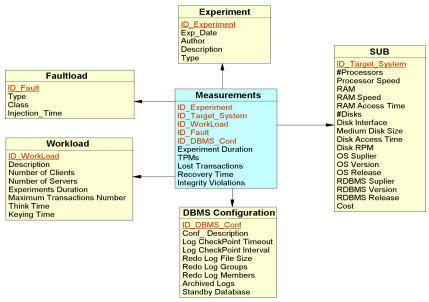
Format of the raw data

- Raw data collected by DBench-OLTP is composed of tens of CSV files (one from each run)
- Each row contains data from an injection slot
  - Identification, duration, number of transactions executed, data integrity errors discovered, type of fault injected, moment of fault injection, workload used, etc)
- A text file describes the experiment and the characteristics of the SUB

Data model (1)

- Key steps:
  - Identification of the facts that characterize the problem under analysis
  - Identification of the dimensions that may influence the facts
  - Definition of the granularity of the data stored in the star schema

Data model (2)



Definition of the loading plans

- Data extraction
  - SQL scripts to extract data from the CSV files to a temporary database schema (data staging area)
- Data transformation
  - SQL scripts transform the data into an adequate format
- Data load
  - SQL scripts to load the transformed data into the data warehouse
- Loading plans documented and stored in the ADR

Load the data

- Executing the loading plans created before
- If new data becomes available we just need to rerun the plans
  - e.g., if the benchmark is executed in other systems
- The documentation of the DBench-OLTP includes papers and technical reports
  - This is considered as part of the DBench-OLTP data
  - It is loaded to the repository and made available to the potential readers of the data

Data ownership policy

- Data ownership policies of ADR are divided in two main groups
  - Private data
  - Proprietary data
  - Collaborative data
- For the DBench-OLTP data we have decided to use a collaborative approach
  - Allows other potential users of the benchmark to compare their results with the ones available in the ADR

Analysis of the data

- On-line Analytical Processing (OLAP) tools
  - Support the analysis in a very flexible way
  - Provide high query performance and easy, intuitive data navigation
- Oracle Business Intelligence Discoverer Plus (ODP)
  - Commercial tool included in Oracle Business Intelligence package
  - Widely used by industry Used freely for research purposes under an Oracle Academy Agreement

OLAP Wizard

- Selection of query type (crosstab or table) and characteristics (title, graph, text area, etc)
- Selection of measures and dimensional attributes
- Setting the query layout
- Selection of the fields to be used to sort the results
- Creation of parameters used to filter data

Some results

	Oracle	PosgreSQL
Tpmc with Faults	1263	644
€tpmc	20	7
Server Unavailability	212	460
Clients Unavailability	13341	23240

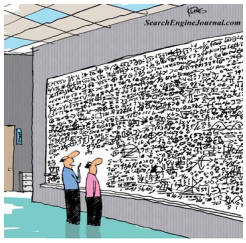
	Integrity Errors
Oracle	0
Microsoft	0
RedHat	0
PosgreSQL	0
RedHat	0

	Server Unavailability		Clients Unavailability	
	Microsoft	RedHat	Microsoft	RedHat
Oracle	194	264	11473	19077
PosgreSQL		460		23240

	Tpmc With Faults	
	Microsoft	RedHat
Oracle	1271	1240

Quick demo…

- Murphy's law… ☺



http://www.amber-project.eu

Do you have data?

Share Them!

Questions

?

Generic bibliography

- Ralph Kimbal, Margy Ross, "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling" (Second Edition), Ed. J. Wiley & Sons, Inc, 2002.
- Ralph Kimbal, "The Data Warehouse Lifecycle Toolkit", Ed. J. Wiley & Sons, Inc, 2001.

ADR bibliography

- Madeira, H., Costa, J., Vieira, M. , "The OLAP and Data Warehousing Approaches for Analysis and Sharing of Results from Dependability Evaluation Experiments", International Conference on Dependable Systems and Networks, DSN-DCC 2003, San Francisco, CA, USA, June 2003
- Pintér, G., Madeira, H., Vieira, M., Pataricza, A., Majzik, I. , "A Data Mining Approach to Identify Key Factors in Dependability Experiments", Fifth European Dependable Computing Conference (EDCC-5), Budapest, Hungary, April 2005

ADR bibliography

- Pintér, G., Madeira, H., Vieira, M., Majzik, I., Pataricza, A. , "Integration of OLAP and Data Mining for Analysis of Results from Dependability Evaluation Experiments", International Journal of Knowledge Management Studies (IJKMS), Volume 2 – Issue 4 – 2008, Inderscience Publishers, July 2008
- Vieira, M., Mendes, N., Durães, J., Madeira, H. , "The AMBER Data Repository", DSN 2008 Workshop on Resilience Assessment and Dependability Benchmarking (DSN-RADB08), Anchorage, Alaska, June 2008
- Vieira, M., Mendes, N., Durães, J. , "A Case Study on Using the AMBER Data Repository for Experimental Data Analysis", SRDS 2008 Workshop on Sharing Field Data and Experiment Measurements on Resilience of Distributed Computing Systems, Naples, Italy, October 2008