Database Tuning Project

Your grade in this class will be based on a written report and an oral exam. The oral exam will be based on your written report and will establish your ability to (a) analyze the performance impact of design choices within a database system and (b) design and execute experiments to evaluate a data management system performance.

The written report is due in three printed copies at ITUs studieadministration on May 25th no later than 3PM. Note that this is a hard deadline. No extension can or will be accepted by the study administration.

Your written report should be composed of two parts.

Part 1

You have the choice between two case studies. You should pick one or the other, but not both. You are welcome to use the results from your 3 assignments as you think is appropriate.

Case A: ReserveWithUs

ReserveWithUs is an electronic portal for hotel rooms. It buys rooms at discounted rates and sells them for a bit more (see hotels.com for a real-life model of such a company).

The company is composed of a sales department that negotiates deals with hotels. Each hotel has a few types of room (e.g., superior, king, queen). For each type of room in a given hotel, the sales department negotiates a block of rooms of a given type at a given price on a given day.

Potential customers browse the web site to find attractive hotels and room deals. Possibly, they log in and add a (number of) rooms to their shopping cart. They might even buy all the rooms in their shopping cart (this is a shopping cart checkout – where the information about the booked rooms are archived).

You are the IT person at ReserveWithUs and you have been given a simple application server. The application server is written in java (tested on java 1.6). The application server together with the database schema and scripts to generate data is available at http://code.google.com/p/databasetuning-cases/ . The design of the app server is described on this web site. The source code is available and you should take some time to familiarize yourself with it. Do not hesitate to contact Philippe if you have questions about the code (e.g., if you are not quite familiar with Java). You will be using DB2 as a back-end database server.

Your job is to tune the application server and DB2 in your environment and to re-design some portions of the application server that are naively designed. Specifically, your project is to document the following tasks (back up your claims with arguments – where possible with quantitative arguments based on experiments):

1. The first issue concerns the architecture. Should the database server and application server be located on the same server? The answer might vary depending on the hardware platform and the software you have at your disposal. You should give arguments for your choice (ideally quantitative arguments).

2. No index is defined in the database. You should pick the appropriate indexes assuming that the workload has the following characteristics (note that you should give arguments for your
choice based on the nature of the workload and on the nature of the queries generated by the application server):

- many queries on hotels and rooms are based on country and city
- many queries are repeated many times (by different users)
- all queries on rooms should contain a date interval
- The sales department loads new deals once a month
- The customers and the hotels grow slowly and regularly (a hundred new customers per month and a couple of new hotels per month)
- There is a lot of activity on the shopping cart (it is required that all information is given when a new item is added to the shopping cart).

3. Once in a while, for maintenance purpose, management requires that you export information about all customers, delete all customers and import the information again. Define a procedure to perform this task as efficiently as you can.

4. You have been told that the checkout function from the application server is slow. You should (a) establish what is the bottleneck (they might be several), and (b) propose a design for a more efficient checkout function. If you have time you can implement this new checkout function and verify that it actually outperforms the existing one.

5. (optional) You have observed that more and more customers search hotels based on their distance to the center of a given city. How can you make such queries as efficient as possible.

6. (very optional) Raise an issue on the project hosting site whenever you find a bug with the AppServer or propose a solution to the existing issues (bugs or requests for new features).

Case B: Your application

You have a case study at work that is not trivial, and where you (a) define one or several interesting tuning issues, (b) document how you use what you have learnt in this class to define where the tuning problem comes from and what a solution might be, and (c) provide quantitative arguments for a solution (and against alternatives).

In your report you should:

1. Describe the context of the application in enough details so that the reader can understand the problems you might run in.

2. For each problem:
   1. Describe the manifestation of the tuning problem(s) that you study (what is wrong?)
   2. Detail your approach to finding what the problem is and present your hypothesis.
   3. Detail your approach to validating/invalidating your hypothesis.
   4. Present your solution and show how it improves on the original situation (make clear what the trade-off is).

If you choose case B, you are very much welcome to check (early on) with Philippe that your case has the potential to give you a passing (or a good) grade.

Part 2

1. You should describe the hardware and OS you have used to deploy the application and
2. You should keep a log of the problems that you face and how you solve them. The goal is twofolds: (a) document the amount of practical work that you had to do, and (b) document the methodology you use to approach and solve problems (it should be based on the scientific method, i.e., (a) formulate one or several hypothesis about the nature of the problem or how to solve it, and (b) experiment to validate/invalidate your hypothesis).