

Database Management and Performance Tuning

Exercises

Pei Li

University of Zurich
Institute of Informatics

Unit 12

Acknowledgements: The slides are provided by Nikolaus Augsten and adapted from “Database Tuning” by Dennis Shasha and Philippe Bonnet.

Exercise 1

- The employee table is on disk 1, the Student table is on disk 2, and the log is on disk 2. The student table is smaller than the Employee table but is accessed more often. Disk 2 supports more than twice the I/O rate of disk 1. The customer is willing to buy a new disk.
- Question: What should you do with it?

Exercise 1

- Solution: Probably the best thing to do is to put the log on the third disk. The log works much better in that case (i.e., as a sequential storage medium), and the system can tolerate the failure of a database disk if there is also a database dump on tape.

Exercise 2

- Response time is quite variable. You learn that new tables are added to the database concurrently with online transactions. Those new tables are irrelevant to the online transactions.
- Question: What you could do to improve response time?

Exercise 2

- Solution: Add those new tables outside the online window. The reason is that any DDL statement will tend to interfere with online transactions that must access table description (catalog) information.

Exercise 3

- A new credit card offers large lines of credit at low interest rates. Setting up a new card customer involves the following three-step transaction:
 - ① Obtain a new customer number from a global counter.
 - ② Ask the customer for certain information, for example, income, mailing address.
 - ③ Install the customer into the customer table.
- Question: The transaction rate cannot support the large insert traffic. Please explain the possible reason and provide a concurrency tuning solution.

Exercise 3

- Solution: The interview with the customer should not take place while holding the lock on the customer number counter. Instead, step 2 should occur first outside a transactional context. Then steps 1 and 3 should be combined into a single transaction. If obtaining the customer number causes lock contention, then obtain it as late as possible, or use a special counter facility such as sequences in Oracle or identity attributes in SQL Server.

Exercise 4

- The business intelligence department would like to run data mining queries on the sales data. While they run their queries, arbitrary updates may occur. This slows down both the updates and the data mining queries.
- Question: What tuning actions you may take?

Exercise 4

- Solution: Partition in time or space. Either run the data mining queries at night or run them on a separate database system.
- You **DO NOT** want to tell people that the payroll will be a day late because someone in marketing decided to see if there is a relationship between 25 independent variables...

Exercise 5

- An application supports thousands of inserts per hour and hundreds of thousands of short update-only transactions per hour. The inserts come packaged as large transactions every 20 minutes or so and last for 5 minutes. When the inserts enter, update response time goes up to 15 to 30 seconds, deadlocks occur, and one of the disks shows exceedingly high utilization. In between the insert bursts, response time is sub-second.
- Question: What tuning actions you may take to help the situation?

Exercise 5

- Solution: The inserts appear to monopolize the system, and the data appears to be poorly partitioned. Two changes in conjunction or in isolation are likely to help.
 - ① Smooth out the insert traffic by chopping the large insert transactions into small ones (if this is possible as far as concurrent correctness is concerned) and issuing them one at a time.
 - ② Repartition the data so that the insert traffic is spread to different disks but the updates still enjoy fast access. A clustering index on a non-sequential key will work well.

Exercise 6

- The system is slow due to excessive processor utilization. An important (relational) transaction executes an SQL query that accesses a single record from within a loop in the programming language.
- Question: What tuning actions you may take to improve the situation?

Exercise 6

- Solution: Replace the loop by a single query that accesses the records and then allows the programming language code to iterate over an array. This will save time because programming language to database interactions are expensive, whereas array-based access is less so.

Exercise 7

- A new credit card company is established. It bills its customers on the last Thursday of each month. The billing transaction takes all night, so other necessary batch jobs cannot be accomplished during this time.
- Question: What should you do to improve the situation?

Exercise 7

- Solution: The first question to ask is whether the application has to work this way. If 1/20 of the bills could be sent out every working day, then the billing application would create fewer demands on the system each day (partitioning in time). Another approach is to run the billing job as a big batch job but only on the weekend.