Adjusting MySQL server

Written by Błażej Pabiszczak on 22 September 2014.

A proper configuration of the database server is a very important element within the system implementation process. This article is not going to describe the configuration procedures but it will highlight a few crucial points in building a server environment. These guidelines are not rigid, they are advices that result from many years of experience within the IT environment:

1. Your friends are: vmstat / dstat, iostat, top, ps and any graphical history of values.
2. Define the bottleneck (read/write, memory, CPU, network). How? Look at the first point.
3. Server optimization according to the process: problem analysis > consideration > change (only one change a time) > test > implement > begin the process again until you achieve satisfactory / optimal results.
4. What could be changed (hardware)
    - RAID 5 is not optimal for databases, it is recommended to use RAID 10.
    - In most cases the speed of CPU is the bottleneck, not the number of cores. It is a good idea to invest in faster CPUs.
    - Use standard 1Gbit everywhere you can (in particular to connect application sever to database server).
    - Use drives with big cache (and protect it appropriately).
    - **Do not use virtualization for database servers!**
5. What could be changed (software)
    - Use optimal systems (SLES, RHEL, Debian, CentOS).
    - Use 64-bit architecture.
    - Use the most up-to-date stable versions (kernel >= 2.6.12).
    - Use mainstream file system, e.g. ext3, xfs.
    - Do not modify your operational system, because it is already optimal!
    - Use a thread cache from your operational system.
6. What could be changed (datebase)
    - The big three (key_buffer_size, innodb_buffer_pool_size, innodb_log_file_size).
    - Do not modify anything, unless you know what it is for. Use ready-made configuration templates, which establish a proper database optimization. Ask a specialist for advise!
    - Unconditionally use InnoDB! (and the optimization for this engine).
    - Monitor slow queries to optimize them.
7. What could be changed (application)
    - Index
        - All attributes where you JOIN.
        - All attributes where you filter (WHERE).
        - All attributes where you ORDER or Group BY.
        - All attributes where you do an Index Scan.
    - How should be indexed
        - Index elements only from left to right.
        - For compound indexes, e.g. INDEX (a, b), INDEX (a,b(10)).
    - The server optimization can increase the application speed to 2x, the application optimization can increase the application operation to 10x.

Print