Performance Features and Improvements since Zabbix 1.8

The following table provides some information about the new features and improvements related to performance since Zabbix 1.8:

Zabbix version	Performance features, improvements, and parameters
Zabbix 1.8	• The configuration data cache module • Frontend images recompressed • SNMP dynamic indexes using one connection
Zabbix 1.8.1	• Fixed the bad performance problem on node synchronization • Support for calculated items (reusing gathered data)
Zabbix 1.8.2	• A new configuration parameter, called `MaxHouseKeeperDelete`, for controlling housekeeping behavior • Frontend sections that use less memory • Node synchronization has received further performance improvements

Zabbix version	Performance features, improvements, and parameters
Zabbix 1.8.3	• Some runtime data is no longer stored in the database • It updates trends in memory and their flushing to disk by caching required information • Faster configuration cache (better SQL queries and in-memory optimizations) • The Zabbix server stores repeated strings (such as item keys) only once in the memory
Zabbix 1.8.4	• Improved network map performance • Trigger `last` and `prev` functions by not retrieving redundant information from the database • Memory usage has been reduced for web monitoring • Reduced memory usage and fragmentation for the configuration cache
Zabbix 1.8.5	• The Zabbix agent's performance was improved, especially on systems with many cores. • The Zabbix daemon's internal process limits were changed. These changes affected the Zabbix server and proxy daemons. • A new internal item was added to monitor the Zabbix process' state. The `zabbix[process,<type>,<mode>,<state>]` item allows us to monitor the busy or idle percentage of different Zabbix server processes over the last minute.
Zabbix 1.8.6	• Improved Zabbix server performance when gathering text data. • Improved Zabbix server performance when processing unsupported items. Previously, information about unsupported parameters was stored directly in the database, bypassing the write cache. Starting from 1.8.6, it uses write cache, thus reducing the database load. • Improved housekeeper performance on PostgreSQL.
Zabbix 1.8.8	• The Zabbix server's performance improved when calculating trigger expressions (now these expressions are cached)

Zabbix version	**Performance features, improvements, and parameters**
Zabbix 1.8.9	• In the frontend, this improved performance for event acknowledging • The Zabbix server's performance was improved in some edge cases by skipping value updating for disabled or removed items • The Zabbix agent's daemon performance on AIX was improved by only collecting data if it was requested
Zabbix 1.8.11	• Unnecessary information, such as user history for nodes and web monitoring items for proxies, is not synchronized anymore
Zabbix 1.8.13	• In the frontend, this improved performance of the **Status of Zabbix** dashboard widget
Zabbix 1.8.14	• This improved screen performance for non-superadmin users
Zabbix 1.8.15	• In the frontend, this improved performance in graph-related pages
Zabbix 1.8.16	• The previously, it was possible to send a large amount of data to the Zabbix server, potentially exhausting the memory. This was now limited to 128 MB when using the Zabbix protocol. Any other data (including older Zabbix protocols) stayed limited at 16 MB.
Zabbix 1.8.17	• Performance of the SNMP normalize function was improved. • Previously, each dynamic index was obtained individually when requested. On systems with thousands of dynamic indexes, Zabbix pollers were completely busy for more than 10 minutes until the cache was built up. From this version onwards, the first dynamic index request would cause all the indexes in the same SNMP table to be looked up and processed in one go. This greatly sped up cache building and reduced the SNMP query count. Note that all entries were looked up even if only some were to be used later.

As we can see, Zabbix SIA has worked very hard to improve the performance of Zabbix 1.8. The most important feature was the configuration and history cache. Since Zabbix 1.8, we have started working with data in a memory area, and the database is not used all the time. Pollers and other internal process have started working without directly accessing the database.

However, Zabbix SIA's efforts to build a great monitoring tool didn't stop there. Let's see what these guys changed in terms of performance in Zabbix version 2.0:

Zabbix version	**Performance features, improvements, and parameters**
Zabbix 2.0	• Multiple sections of frontend had their performance improved. • Improved housekeeper performance on PostgreSQL. • Caching of general configuration (unsupported items, the discovery group, custom severity names, alerts, and events). • Trigger cache was introduced (it holds a description, an expression, an error, a severity, a type, a value, and `value_flags`). This resulted in improved performance when processing events. • Improvements in user macro caching. • Improved history (DB) syncer and escalator performance.
Zabbix 2.0.1	• Previously, the Zabbix agents allocated a static amount of memory to monitor eight disks or partitions. This amount now became dynamic and was allocated only if monitoring of a specific disk or a partition was requested. This reduced memory usage on systems where fewer than eight disks were being monitored.
Zabbix 2.0.2	• In the frontend, the performance of graph-related pages was improved • In the frontend, the performance of the dashboard's **Host status** widget was improved, especially on setups with many groups
Zabbix 2.0.3	• Performance of the screen, host screen, and graph sections was improved
Zabbix 2.0.4	• In the frontend, page refreshing would now reload less files, thus improving overall frontend performance and decreasing network traffic

Zabbix version	Performance features, improvements, and parameters
Zabbix 2.0.5	• In the frontend, overall performance was improved by optimizing performance checks • In the frontend, the SQL query count was reduced, and performance was improved in IT services and most pages that accessed trigger information • Performance of Oracle as the Zabbix backend database was improved by prefetching 2 MB of data in `SELECT` statements • Improved SNMP performance • Increased cache size limits (`CacheSize`, `HistoryCacheSize`, `TrendCacheSize`, and `HistoryTextCacheSize`) to 2 GB
Zabbix 2.0.6	• Previously, low-level discovery could generate a large number of database queries, and even result in deadlocks. Starting from Zabbix 2.0.6, each low-level discovery prototype would be processed in its own transaction, avoiding the deadlocks. • Already discovered items would be updated only if some of their properties had changed. • Only changed fields would be updated. • The amount and size of the SQL statements was reduced. • The amount of traffic exchanged between the Zabbix server and the Java gateway was reduced.

Zabbix version	Performance features, improvements, and parameters
Zabbix 2.0.7	• Performance of the dashboard's **System status** widget was improved by an average of 3-7 percent for both memory usage and execution time. • The time required on 100,000 unsorted values was reduced from 133 seconds to 2.5 seconds. This low-level improvement was likely to help with frontend performance on large installations. • Oracle: Zabbix 2.0.5 improved performance by adding a 2 MB prefetch. In some cases, this could have led to worse performance. Thus, in Zabbix 2.0.7, the prefetch was changed to be row-based, which would improve the performance in those edge cases. • Previously, Zabbix used a configuration cache mutex and a string pool mutex. Starting from Zabbix 2.0.7, only one mutex was used. This change reduced the amount of time needed for the initial update of the configuration cache by an average of 20 percent. • The Zabbix proxy performance was improved by reducing the number of database queries on the proxy side during configuration updates. • No invalid recovery escalations. This, in the case of big installations, could have affected the performance of the database. • Improved performance of the `zabbix[queue]` internal check.
Zabbix 2.0.9	• Loading graphics was much improved by faster working SQL statements, significantly (up to 100-1000x) improving the loading times with all databases.
Zabbix 2.0.10	• The loading speed of several screens of the **Monitoring** section (**Last 20 issues**, **Latest data**, **Status of triggers**, **Latest events**, and so on) with a large number of global scripts was improved

Zabbix version	**Performance features, improvements, and parameters**
Zabbix 2.0.11	• Trigger processing performance during low level discovery improved. • A trigger could now only be processed by one main history syncer or timer process at a time. They would not do duplicate work by processing triggers that were already being processed by history syncers.
Zabbix 2.0.12	• Graph processing performance during low-level discovery was significantly improved. • Batch processing of IT services was added. It resolved possible deadlocks and improved performance when processing large IT service trees (a 300 percent performance improvement).

As we can see, in Zabbix version 2.0, Zabbix SIA's guys improved the cache and buffer functions. Another visible effort was improvement of the Zabbix frontend. The maximum cache size changed to a bigger value. Here, the trigger cache was the most important performance feature. This feature is part of the configuration cache and holds information about triggers (description, expression, severity, type, and so on). It improved performance with triggers and reduced the load on the database.

What about Zabbix 2.2? Have we had any performance improvements? Let's check:

Zabbix version	Performance features, improvements, and parameters
Zabbix 2.2	• Loadable modules, a way of extending the Zabbix functionality that is more performance-minded than the user parameter option or external checks. • Allows the monitoring of proxy performance metrics. • To reduce the number of update operations, shared memory was now used to store fields related to the last and previous value of items, causing great improvements in the server's performance. • The Zabbix server and proxy daemons would now support bulk access to configuration and history caches. This would reduce the quantity of system calls of operation with semaphores and positively affect the system's performance. • The Zabbix pinger processes would not use a connection to the database anymore. • Redundant indexes were removed from several of Zabbix's MySQL database tables. This was likely to improve performance and slightly reduce the database size of MySQL. • Such indexes used to be created automatically on MySQL. Now they were also created on PostgreSQL, Oracle, and DB2 to improve the Zabbix server's performance. • `ValueCache` was introduced. This in-memory cache could be used to access historical data, instead of making direct SQL calls to the database. If historical values were not present in the cache, the missing values were requested from the database and the cache was updated accordingly.
Zabbix 2.2.2	• The `ValueCache` memory efficiency improved. Now it required less shared memory to cache the same number of values. • Trigger processing performance during low-level discovery was improved.

Zabbix 2.2.3	• SNMP monitoring performance was significantly improved by introducing bulk requests with a maximum of 128 items. The load on the Zabbix server and monitored SNMP devices was likely to be greatly reduced. • Graph processing performance during low-level discovery was significantly improved. • On Oracle databases, variable binding was now used for bulk inserts, resulting in much better performance.
Zabbix 2.2.4	• Only values that fell within the last 24 hours were now displayed in **Monitoring | Latest data**, **Monitoring | Overview**. This limit was introduced with the aim of improving initial loading times for large pages of the latest data. • History cache performance was improved by using the configuration cache more, instead of using the database. • Trigger-based events were loaded much faster and with less memory usage in **Monitoring | Events**. • Last event calculation in the **System status** frontend widget was optimized, which may have resulted in improved speed and less memory usage in environments with a large number of problematic triggers and host groups.
Zabbix 2.2.6	• The performance of last value retrieval by the `item.get` method was improved to use values from 24 hours only (by default).
Zabbix 2.2.7	• Value cache requests were optimized to better utilize database indexes. The improvement would be mostly noticeable with large databases.

In Zabbix 2.2, the `ValueCache` feature was introduced. This feature would improve performance with trigger expressions, calculated and aggregated items, and some macros. With `ValueCache`, the Zabbix server would access historical data in the memory instead of making direct SQL calls to the database.

In Zabbix 2.4, we got some more performance-related features:

Zabbix version	Performance features, improvements, and parameters
Zabbix 2.4	• On startup of `zabbix_server` and `zabbix_proxy`, housekeeping is postponed for 30 minutes instead of running at once. This will lower the startup load for both of these processes. • The formatting of JSON objects (with tabs and new lines) has been removed, which allows the traffic to be reduced by 20-30 percent when data is sent between Zabbix services. Additionally, the escaping of the forward slash, or solidus (`/`), has also been removed. • Improved performance of the deletion of triggers by the API and server.
Zabbix 2.4.1	• Value cache requests have been optimized to better utilize database indexes. The improvement is mostly noticeable with large databases.

Even now, Zabbix SIA is working to improve Zabbix's performance. The changes to the JSON formatting will save network traffic to monitoring functions up to 30 percent.