

Blackboard Help

Learn / Administrator / Self- and Managed-Hosting / Release Notes / Individual Releases / 9.1 SP 10 / Performance and Scalability

Performance Improvements

As part of the release, performance optimizations are introduced back into the product as a result of a customer issue or internally found problem. The Blackboard Performance Engineering team is responsible for refactoring under-performing areas of the application and verifying regression improvements based on the optimization.

System

High amount of physical reads from disk caused by "where" clause in sync_activity_accumulator stored procedure

A fix was implemented to add a condition that avoids copying of unnecessary records in the procedure.

Need improved search indexes throughout the product

Unnecessary database indexes were removed and appropriate ones have been added.

VTBE symbol font parsing causes high CPU utilization due to inefficient regular expressions

A regular expression pattern has been refactored to optimize the symbol parsing process.

Thousands of "Timer-xxxxx" being leaked causing descriptor exhaustion and server lock-up

A timer thread leak in CAS authentication provider was causing the issue. The timer threads are now properly managed.

CPU spike due to ISAPI_REDIRECT.DLL file

Tomcat connector libraries have been updated to eliminate the CPU spikes when serving large files. The fix addresses the problem on the supported 64-bit platforms only.

Email file attachment uploader can cause high CPU usage. This can be used as a DOS attack.

Cross-site scripting filter's regular expressions were optimized to mitigate the issue.

Updater: Some gradebook post_update_sql scripts repeat unnecessarily

Certain post upgrade scripts within the installer used to run multiple times unnecessarily, causing system

upgrades to take longer than they should.

HTTP Compression disabled by default

Web server compression is now enabled by default. However, when you enable SSL on the default version of Apache 1.3, HTTP compression will no longer work. In order to use both SSL and HTTP compression, you must install Apache 2.2.

Tomcat shutdown slow on application servers not running the ActiveMQ broker due to file lock contention

Tomcat shutdown process no longer tries to stop the ActiveMQ broker if it isn't actually running on the current node; trying to do so causes a deadlock.

High reflection overhead while performing database object unmarshalling

A pattern of CPU utilization due to reflection overhead during database column value unmarshalling was detected and fixed.

Course Delivery

UserMsgStateDbLoaderImpl and LoadFromCourseIdQuery have a bug and performance issues

Database index changes were introduced to fix the performance issue.

msg_parent_position() and msg_rating() are not performing well with large forums

New indexes have been added to reduce the database "consistent gets" for improving the performance of msg_rating and msg_parent_position functions.

During course archive, the Content Alignments process takes 23 hours to complete

Underlying API changes have been made to retrieve alignable contents more efficiently.

Overall summary of usage report query expensive across core tables (users, course_users, activity_accumulator tables, navigation_item and application)

The issue has been fixed via database query optimizations.

Reoccurring QTI table deadlocks

Persistence logic of the assessment objects was refactored for efficiency and elimination of deadlock on qti_result_data table.

Content Collection

High Learn session counts due to Mac OS WebDAV client behavior

Finder WebDav client does not send back cookie in the follow-on requests on Mac OS. This prevents maintaining a session and also generates lots of unnecessary sessions on the server side. Session data has been disabled for Mac OS WebDAV until Apple resolves the issue.

Long running XYF_FULL_TEXT_SEARCH_TEMP queries and problematic index hints

Database query optimizations and addition of new indexes have been made to fix the issue.

360 view search results are exhausting heap

Eliminated the Files and Subfolders section of the 360 view for top-level folders only.

Recheck Content process performs poorly

Refactored the implementation logic to reduce fetching, loading, traversing, and committing of data.

Infinite Recursive Loop Detected in DirectoryEntry causing consumption of Java heap while writing to logs

Context release was not being triggered upon exception. Better error handling has been implemented.

Delayed connection release in SlideShareApiImpl

The method for releasing HTTP connection was not getting called explicitly. This was causing the HTTP client to hold on to the connection 100 seconds longer than necessary.

Xyθος directory quota lookup executes full table scans and causes DB row lock contention

A database query has been tuned in order to maintain its query plan stability against various data sets.

Community Engagement

Massive number of active DB sessions stuck on slow system_roles_entitlement CONNECT BY PRIOR query

Refactored the UI to eliminate loading of viewable multi-institutions hierarchical nodes unless the user explicitly requests a specific node.

Slow user search against domains in 9.1 SP8

It was caused by an inefficient query that was fixed by the use of EXISTS instead of IN function.

9.1SP8 System Admin courses search results taking 60-500 seconds LRN-51725 to load

The issue has been resolved by index changes and database search query optimizations along with the

support for CTE queries in the pagination framework.

DB performance tuning around LAYOUT_FAMILY processing and new index on LAYOUT_FAMILY(PK1,TAB_PK1)

Index optimization change has been implemented to resolve the performance issue.

Notification query (enrollments again) causing massive system load spike

New indexes have been added to reduce the database "consistent gets" for improving the query performance.

Setting System Admin as secondary role in a system with domains results in high heap usage

The node entitlement retrieval code was looking at the default domain when there is no need to consider what's inside the default domain.

Poor performance from LoadInstructorsByUserQuery query

The issue has been fixed via database query optimizations and new indexes.

Portal Modules That Are Dependent On External Services Can Cause Tab Page To Load Slowly

A module that relies on external service can slow down an entire page if it's loaded synchronously. As of SP10, modules are configured to load asynchronously by default. Administrators or B2 developers can change this default behavior.