Performance Tuning Reference

Planon Software Suite

Version: L111

Planon
Building Connections

About this Document

Intended Audience

This document is intended for *Planon Software Suite* users.

Contacting us

If you have any comments or questions regarding this document, please send them to: support@planonsoftware.com.

Document Conventions

Bold
Names of menus, options, tabs, fields and buttons are displayed in bold type.

Italic text
Application names are displayed in italics.

CAPITALS
Names of keys are displayed in upper case.

Special symbols

	Text preceded by this symbol references additional information or a tip.
	Text preceded by this symbol is intended to alert users about consequences if they carry out a particular action in Planon.

Table of Contents

Introduction

This document outlines the key factors that influence the performance of Planon applications. It provides an overview of what application performance entails and how it can be optimized.

Some performance enhancements can be implemented by customers, while others require action from Planon. We will explore these possibilities in detail. The information provided serves as a guideline and starting point for performance tuning.

Planon strives to build software solutions that respond fast. Focus on performance is part of the DNA of our development teams. This document outlines our approach. We aim for transparency in our operations to assure you that updates to new features or versions are reliable.

While we are confident in our robust methodology, we acknowledge that no software is completely free of bugs, and performance issues may still occur. In such cases, our support center is ready to provide solutions. This document serves also as a guide to help you analyze issues and assists you in providing the necessary information to our support consultant.

Scope

The scope of this document is Planon Universe. Products developed by other Planon subsidiaries, like Control IT's Bisonbox or Reasult's platform, are not covered. However, many sections of this document can universally be applied to other solutions as well.

What is performance?

What do we mean when talking about the performance of an application?

The most crucial aspect of performance is speed, which refers to how quickly the software responds to user inputs, processes data, and displays results. This is often indicated as response time, and it is the aspect that the end user directly experiences.

However, there are other aspects that come into the picture. Performance can be divided into several categories. Another such category is scalability. This refers to the software's ability to maintain or improve performance as the load increases. This increase in load can be caused by an increase in the number of users, number of concurrent tasks, or data volume.

The above two performance aspects, speed and scalability are related to the end-user experience. There are also aspects that are more of a technical nature.

The first one is throughput. Throughput is the number of tasks or requests that the software can handle simultaneously or in a given timeframe. This is often referred to as transactions per second.

Another one is resource usage; this indicates how efficiently the software uses system resources, such as CPU, memory, disk I/O, and network bandwidth. These technical aspects are not directly linked to the experience of end users but can be used by development teams to tweak the performance.

What determines performance?

This chapter explains all the factors that influence performance. These factors can be split into multiple categories, such as hardware, software, configuration, data, and workload.

Hardware

The capabilities of the hardware on which the application is running can greatly affect the performance.

Memory

Planon runs on a Java Virtual Machine (JVM). The initial and maximum memory that will be allocated to a JVM can be controlled in configuration files. The allocated memory is called heap space.

The total amount of maximum heap space on a system should be less than the memory allocated to the system. Sufficient memory should remain for the operating system and potentially other applications.

JVMs make use of a garbage collector to free up memory. The garbage collector runs occasionally, or when memory is needed. The JVM controls when collection is done. (Please note that the Windows Task Manager only displays the allocated heap size, not the heap size used.)

The Planon web server JVM has a more dynamic nature of memory usage than the application server JVM. This causes the web server JVM to benefit more from additional memory than the application server JVM. Contrary to the application server, to grow the number of users on the web server, we need to increase its assigned heap size.

CPU

A CPU (Central Processing Unit) is the primary component of a computer that performs most of the processing inside the system, executing instructions from programs and managing tasks. The performance of an application is significantly influenced by the CPU's speed and efficiency, as a faster CPU can execute more instructions per second, reducing the time required for computations and improving overall application responsiveness. The CPU usage of the Planon application server, web server, and database is linear and scalable with the load.

Disk I/O

Disk I/O (Input/Output) refers to the read and write operations performed by the storage devices, such as hard drives or SSDs, within a computer system. Disk performance is not considered to have a major impact on the performance with respect to the application server and web server.

Network I/O

Network I/O (Input/Output) characteristics refer to the performance attributes of a network, such as bandwidth, latency, and throughput, which determine the speed and efficiency of data transfer between systems.

When the application server and database are located on different servers, we recommend not only to consider throughput but also latency. Also consider network characteristics like (reverse) proxies, firewalls, and communication protocols (HTTP(s), RMI).

GUI client

End users communicate with the Planon Universe system via a graphical user interface (GUI), accessible via a browser on a laptop or a smartphone app. The GUI communicates via the network/Internet with the Planon backend systems, where most business logic calculations are executed.

The client’s system is responsible for displaying the content received from the backend system. The hardware, network, and software capabilities of the end user’s system impact the speeds at which the content can be rendered. The impact of the system’s capabilities is especially seen when a lot of data needs to be displayed, such as complex CAD or BIM drawings.

Software Applications and Settings

Database indexes

Database indexes are data structures that improve the speed of data retrieval operations on a database table by allowing quick access to rows. They relate to the performance of an application by significantly reducing the time required to query and fetch data, thereby enhancing the application's responsiveness and efficiency, especially for large datasets.

Database statistics and query optimizer

Database statistics are metadata that describe the distribution and storage characteristics of the data within a database, such as the number of rows, data distribution, and index usage. They relate to the performance of an application by enabling the database management system to optimize query execution plans, leading to faster and more efficient data retrieval and processing. Statistics should be updated regularly to provide the query optimizer with accurate information.

Database characteristics

Database engineers can tweak a wide range of characteristics of a database server. Some examples are the minimal and maximal server configuration, the max degree of parallelism (MAXDOP), processor affinity, tempDB configuration, cache plan optimization, and many more. The database server brand and version also influence performance. A database server can be used by multiple applications/customers or one. The performance of a shared database server can be impacted by the load that is placed on it by other applications or customers. For example, many default Planon SAAS cloud environments share the same database server.

Compression

Compression is the process of reducing the size of data by encoding it more efficiently. It relates to the performance of an application by decreasing the amount of data that needs to be stored or transmitted, which can speed up data transfer rates and reduce storage requirements, ultimately enhancing the application's performance.

Operating system

The operation system that hosts the application impacts performance. For example, Linux is seen as faster and less resource-intensive than Windows. The application and web servers of our SAAS cloud environments run on a Linux distribution. The operating system for our database servers remains Windows.

Other software

Other software running on the systems hosting Planon can also influence the performance, such as virus scanners or maintenance tools.

Planon software version

The design and implementation of our software itself play a crucial role. We strive to optimize our performance as much as possible. Many optimizations are continuously considered and applied, such as caching, database indexes, stored procedures, query optimization, and data normalization. See the chapter on this topic for an explanation of how we do this.

Software Configuration

Another major factor that determines the performance is the way the Planon application is configured.

Think about context filters, drill-down paths, and authorization association links. Queries to retrieve data from the database will depend on such configurations.

Another example is the use of additional code that could be executed, such as additional business rules via TMS or PaaP apps.

Data volume

Data is stored in database tables. The more data there is in the database table, the longer queries will take. This can be partly overcome by applying indexes. Indexes can greatly increase the performance of *read* queries but will slow down *create*, *update*, and *delete* requests.

Workload

Last, but not least, the system performance is greatly related to the number of concurrent actions on the system.

- User actions:
  - Concurrent users: the number of users simultaneously communicating with the system.
  - Think time of users: The lower the think time, the faster the users click through the application, and the more communication the system must process.
- Automated actions:
  - Background actions: Actions can run in the background, for example, scheduled daily processing of certain data.
  - Alerts.

- Communication triggered by other systems (for example Kiosk, Room booking Panel, Exchange, SalesForce, SAP, IoT sensor input, SDK usage, Webservices usage).

Conclusion

The performance of an application depends on many aspects like hardware, software, and configuration.

The actual performance is always determined by the number of tasks that need to be processed (determined by software, volume, and load) and the slowest component (determined by hardware).

How can customers tweak performance?

We have seen many factors that influence performance. Some of these factors are simply given and cannot be tweaked by customers, but some can. This chapter discusses the possibilities.

Cloud and on-premise

What is available to you also depends on the service model, SAAS, or on-premise.

If you are using Planon SAAS, the backend runs in a cloud infrastructure (like Amazon AWS or Microsoft Azure). The Planon Ops team ensures that your environment is provided with sufficient hardware capabilities, such as CPU and RAM, and adequate infrastructure settings.

For example, the Ops team has options to assign more hardware, implement multiple servers, and use a dedicated instead of a shared (database) server. SAAS customers do not have the ability to tune these infrastructure and hardware capabilities, nor do they need to do so.

This section discusses options that are available for both models.

In a Planon Cloud environment, performance can be improved by creating custom indexes on the database. In a cloud environment, you can create database indexes in Field Definer, with the help of Planon support. This saves you the trouble of involving a database specialist to manually add them to the database. See the online WebHelp for more info. In an on-premise environment, the customer's DBA can create indexes.

Software Configuration

This section describes many options that the administrator can use to influence performance.

Drill down selection

Although it is possible to select twenty thousand records and drill down to their details, the performance will be poor. The admin can help the end users by creating appropriate filters and step filters.

Records per page

You can specify the maximum number of elements that are displayed in the elements list. The lower the number, the faster the list is displayed. The maximum number is five hundred [WebHelp link].

Scheduled processes

When a scheduled process is running, it will consume system resources, which could result in decreased response times for end users. If possible, it is advised to run scheduled jobs outside of business hours.

Generating reports

Some reports are generated by processing a lot of data. Such reports consume considerable amounts of application server memory. If possible, it is advised to execute such reports outside of business hours.

Polling frequency

Many background processes poll data from other sources at a certain interval. For example, cost centers can be imported from SAP. Another example is the polling frequency for measurement points (IoT sensors). Consider if such frequencies are not higher than required.

Import/export chunks

Import-/export documents are processed in chunks. You can define the size of these chunks by specifying the Fetch size. Using chunks can prevent transaction time-outs. The chunks will be processed as separate transactions within the same run. This feature applies to both manual and scheduled runs [WebHelp link].

Authorization association links

When two business objects are linked using an m-to-n relation, the second business object (N, on which the Authorization filter is created) can be made available as

an Association link to the first (M). Although this is a powerful mechanism, with authorization links on reference fields and on associations it is possible to go two directions and create long chains. These chains are complex and slow queries. The advice is to be conservative when using association links [WebHelp link].

Mail merges

When generating mail merges, consider the size of the included content. If each mail is enriched with multiple images of several MB, the performance will be slower compared to a mail merge of a small size.

File storage

The file pop-ups can display a lot of details of files (like size, and date). The more files are in one directory, the slower the performance. Consider if a more fine-grained directory structure could be implemented. You can automatically create a logical file structure on the file system, by enabling the *Planon managed* functionality [WebHelp link].

Business Analytics usage

For large reports, where the information does not need to be processed in real-time, it might be better to use our data lake solution with MongoDB. Planon's Connect for Analytics extends the operational Planon system with a data lake and a data connector to external data analytics tools. This simplifies the extraction of data for reporting, analytics, and predictions with standard BI tools [WebHelp link].

CAD Integrator

For large CAD drawings, where all the details in construction drawings are not required, it might be better to consider the 'Level of detail' field in the 'Floor attributes' business object [WebHelp link].

Additionally, an automated system is in place that converts larger .svg construction drawings to image format, enabling faster loading times for CAD viewer drawings [WebHelp link]. Currently, space administrators cannot restrict this conversion. However, starting with release L110, there will be a setting in 'Floor attributes' BO that allows administrators to choose whether to proceed with this automatic conversion or not.

Automatic conversion will not be applied to Kiosk.

Data volume

The larger a database table, the slower a query. Consider if you could delete old unused records. (If you need them for future reference, you might export them or store a database backup). An option is to clean certain tables via a scheduled task. This approach is implemented by default for certain records (for example, old session data, old event logs, old tasks, processed inbound and outbound messages) [WebHelp link].

Event connector

Messages that get stuck are interrupted (since L106) given the timeout configured in System Settings. For best scalability, this timeout should be set as low as possible. The performance logs can be used to get information on the timing of Event connector solutions [WebHelp link].

On-premise only

In addition to the previous options, on-premise customers have further options for tuning performance.

Hardware capabilities

The hardware resources play a key role in performance. Customers who maintain the Planon app and web server within their own setup can influence these resources. This includes running the software on systems with sufficient CPU, RAM, disk I/O, network throughput and latency, database server tuning, etcetera. As these are not specific to Planon, we will not further elaborate on them.

Memory heap size

Sufficient memory needs to be allocated for the Java processes. Sufficient RAM needs to be available to prevent memory swapping (virtual memory usage). For details see the Administrator’s Guide. Note that more memory does not necessarily result in better performance. Increasing the heap sizes will only increase performance if the memory usage is a bottleneck. A bigger heap size could result in a higher CPU load, resulting in a lower performance.

Database statistics

Database statistics must be periodically generated and updated on all the tables and indexes, to ensure that your database is using the optimal path to access data.

Multiple application and web servers

Multiple application servers and the web can run in parallel to spread the load. For more information, refer to the Deployment Guide (Application server clustering) or the Installation Guide (Runing the suite installer, Step 12).

How does Planon focus on performance?

Introduction

Focus on performance is an integral part of our software development life cycle. All development teams are required to consider performance from the inception of an idea through all subsequent steps including production.

This involves defining performance requirements, designing and coding with performance in mind, and preventing regression. Several tools are available to our teams, like documented coding standards, a best practices cookbook, and a support team with performance experts. Our QA approach is risk-based.

Technical and business risks are assessed for any code change. If a performance regression risk is identified, the team decides how to mitigate it, for example by creating an automated test. Automated performance regression tests run daily on the latest build. These tests run on dedicated infrastructure so that the results do not suffer from variations in CPU, memory resources, network bandwidth, and the like. If a test detects a deterioration in performance, the responsible team will analyze and take appropriate action.

Requirements phase

When initiating a project, introducing a new feature, or exploring a customer idea, taking into account the performance aspects of making such changes is considered essential. Performance can play a big part in a customer’s experience and use of a system.

When considering a new solution, both functional aspects and performance requirements are explicitly defined. These requirements encompass not only the system's response time or the speed at which it returns a value but also account for the expected volume and size of the dataset deemed ‘normal’ usage by the customer during development.

Performance requirements and risks are discussed in collaborative sessions, involving representatives from development, testing, and typically the product owners. These sessions serve as a pivotal platform for aligning understanding and expectations across teams.

Design phase

After establishing the performance requirements and risks, the development team creates functional and technical designs. The outcome of the previous phase allows the

team to collectively evaluate the technical implications of these requirements. Devising strategies to optimize performance without compromising functionality.

By openly addressing performance requirements within our “three amigos” sessions, teams can work to establish thresholds and limits that serve as guardrails against potential performance pitfalls.

These limits help prevent inadvertent design decisions or implementations that could lead to scalability issues or performance bottlenecks down the line. The technical designs are reviewed by our architects.

Coding phase

Once development starts, it is considered crucial for all developers to conscientiously consider the performance implications of the code they write. Where feasible, developers proactively implement (agreed) limits within the system to pre-empt any potential future performance bottlenecks.

To help aid developers in creating performant code, performance engineering guidelines are available to them. In addition, developers can apply specific tools to measure and inspect code performance. When a developer needs to have his code interact with the database, methods of a specialized database toolbox can be used.

This toolbox ensures high-quality code that adheres to our coding standards. Once the code is created, it is subject to peer review. If new code for database queries needs to be written, the code is also reviewed by a database administrator (DBA).

Testing phase

The testing phase extends beyond just ensuring the functionality of the software meets the required standards; it also includes validating its performance. Performance testing is incorporated into our testing strategies, as it is essential for delivering software that not only works but also performs optimally.

All teams have had extensive training in performance engineering and testing. When a performance risk is identified, the team will define an approach to tackle this risk. An entire range of techniques and tools are available.

To ensure that newer versions of our software stack do not result in deteriorated performance, we have a set of performance regression tests that run daily on the latest code base. Only after all the risks are mitigated, is a team allowed to release new software into our base trunk.

Production phase

Performance depends on many factors that are comparable between customer environments, like the hardware and infrastructure, while other factors vary a lot, like all functional configurations and the amount of database records.

As Planon is highly configurable, the number of possibilities is endless. We will never be able to predict, let alone test all possible configurations. Performance issues in production cannot be prevented 100%. We continuously monitor the availability and performance of our cloud environments and take proactive precautions.

I have a performance issue, and now?

If the performance of Planon software solutions is inadequate, our support team is ready to investigate the issue and provide a solution.

During this investigation, we will have to find the bottleneck together. To help our team swiftly debug the situation, you are encouraged to gather this information up front:

Question	Explanation
What exactly is slow?	If it is only slow at a specific moment please provide us with this timeframe. Also, think about whether there was some scheduled job running at that specific time or if there were any changes made to the system.
Is the issue reproducible?	And if so, is it reproducible on other environments (ACC, TEST, DEV, PROD)? (To check if the performance issue is still present, is it always there or only when some other process is running? If the issues are reproducible, this is something we can focus on and test in a demo environment).
Are there any errors shown?	If so, please provide us with a clear screenshot of the error. (This will help us correlate the error with the log files).
Is the whole application slow, or only specific parts?	To pinpoint the issue and exclude probable causes where the issue originates.
Are there other applications that are slow as well?	Example: How does an application like YouTube perform? (This will help determine if there is slowness in multiple applications that are not related to Planon. If that is the case, you need to get your own IT department involved to manage this.)

Question	Explanation
When did the performance issue start?	What time? What changed? Is it constant? (This information is needed to identify what possible changes/situations could have led to this issue and to know where to start analyzing and excluding)
Do multiple users experience these performance issues?	To find out if there are any similarities between the users, regarding user group, authorization, or (infrastructure of) location.
Is it currently still slow as we speak?	If not, can you relate it to any action that happened at the time of the slowness?
Is performance monitoring enabled?	To determine if the problem is related to the application server or database server or any other component.
Is Tailor Made Software being used?	If Yes, on which business objects? (In the case that TMS is involved this can affect the performance in your environment. If possible, test without TMS enabled, for instance, on a test environment.)
Are scheduled tasks/imports/web services running?	If a heavy action is running, this can negatively affect the multiple resources of Planon. Think of Imports, web services, or even large reports that are running.
For on-premise installations: Have any adjustments been made to the infrastructure?	For this please think of your IT department doing any maintenance to either the company network or implementing patches on the server.
Log files	In addition to the above and to analyze the performance issue further, please provide us with the following log files. • Wildfly (Wildfly application server log) • Tomcat (Tomcat web server log)

Question	Explanation
	• Localhost access (http requests to tomcat log) • Performance (default performance monitoring log) • Performance Extended (extended performance monitoring log) (if enabled)

Index

P

T