Teradata Performance Tuning

Workload Management with TIWM and TASM

If you want to support our work you can buy the Teradata Performance Tuning Book I wrote together with Artemios Vogiatzis at Amazon:

amazon

Table of contents

Introduction

To all friends of www.dwhpro.com, your loyalty, and dedication as readers of our blog posts have been the driving force behind our continued efforts to share valuable insights and knowledge. We extend our heartfelt gratitude and dedicate this pdf file to all of you.

Chapter 5: Teradata Workload Management

5.1. What is Teradata Workload?

Teradata workloads are an essential aspect of managing database performance and resource allocation. They consist of collections of requests that share similar attributes, such as query type, user group, or application. By grouping these requests, database administrators can more effectively categorize, monitor, and analyze the utilization of system resources, such as CPU, memory, and I/O.

Each query is always assigned to a specific workload based on the classification criteria defined for that workload. The classification criteria consider the query's characteristics and the login information associated with the session to which the query belongs.

For instance, if a query matches the classification criteria of a workload in Timeshare TOP, it will be executed within that workload. Similarly, if the query matches the classification criteria of a workload in an SLG tier, it will run within that workload. This way, workload management effectively determines the appropriate workload for each query, ensuring optimal resource allocation and system performance.

If a given query could potentially map to two workloads, one with classification based on user and another on account, the Workload Evaluation Order in TASM will determine which workloads the query will utilize. When finding a suitable workload for a query, the workload management assesses all possible workloads in the Workload Evaluation Order. It then matches the query to the first workload where the classification criteria are met.

Why are we grouping requests?

Grouping requests into workloads offers several benefits. First, it allows for identifying trends and patterns in resource consumption, helping to pinpoint inefficiencies or areas that require optimization. Second, it enables more granular control over resource allocation, as administrators can manage and prioritize resources at the workload level. This ensures that resource usage adheres to established guidelines regarding the quantity, duration, and frequency of resources allocated to each specific group of requests.

Additionally, the workload management concept in Teradata helps maintain a balanced and efficient resource distribution throughout the system. Administrators can ensure that high-priority workloads are processed on time by allocating resources based on priority and business requirements. In contrast, lower-priority workloads are handled without monopolizing system resources. This ensures the system remains responsive and adaptable to varying demands and promotes stability.

Furthermore, understanding and managing workloads can lead to better capacity planning and resource allocation decisions. By monitoring the performance and resource usage of different workloads over time, administrators can make informed decisions about hardware upgrades, system tuning, or architectural changes. This, in turn, helps maintain optimal system performance and scalability, ensuring that the database can continue to meet the organization's evolving needs.

In summary, Teradata workloads provide a framework for organizing, managing, and optimizing the allocation of system resources. By grouping requests with similar characteristics, administrators can more effectively monitor resource usage, prioritize tasks, and maintain a balanced distribution of resources throughout the system. This ultimately results in improved performance, stability, and adaptability of the database, ensuring that it remains capable of meeting the growing and changing demands of the organization.

5.2. Introduction to Teradata Workload Management

Teradata Workload Management (TWM) is a suite of software tools and services designed to optimize the performance of Teradata Database systems by effectively managing workloads we defined as described in the previous section. It ensures the system performs at its peak, delivering optimal query response times and throughput while maintaining system stability and reliability.

TWM balances the competing demands of various users, applications, and workloads. It intelligently prioritizes and allocates system resources to different workloads, ensuring critical business processes receive the necessary resources. In contrast, less critical tasks are executed without causing performance bottlenecks or overloading the system.

5.3 Teradata Integrated Workload Management vs. Teradata Active System Management

Teradata Integrated Workload Management (TIWM) and Teradata Active System Management (TASM) are both solutions offered by Teradata to optimize system performance and manage workloads in a Teradata Database environment. Although they share some similarities, they have key differences in their functionality and purpose.

TIWM is a built-in feature of the Teradata Database, included in the base software. It provides static workload management capabilities that help organizations manage their resources and workloads.

TIWM offers workload classification, prioritization, and resource allocation based on query characteristics, user-defined attributes, and system-defined attributes.

It enables administrators to define workload management rules and policies, balancing resource usage among various users and applications.

TIWM is suitable for smaller-scale Teradata environments where basic workload management capabilities are required and resource contention is minimal.

TASM is a more advanced, optional workload management solution designed for larger, complex Teradata Database environments with more demanding workload management requirements.

Teradata Integrated Workload Management (TIWM) utilizes static workload definitions, whereas Teradata Adaptive System Management (TASM) enables dynamic adjustment of workload management based on actual system conditions. This distinction allows for a more adaptable and responsive approach to managing workloads in TASM.

In TIWM, workload definitions are fixed and do not change in response to varying system conditions. This means the same rules and priorities are applied consistently, regardless of system state or resource availability.

On the other hand, TASM allows for dynamic adjustments to workload management depending on real-time system conditions. These adjustments can be triggered by time-based events, such as reaching a specific time of day, or system-specific events, like exceeding a predefined number of AMP Worker Tasks (AWT) in use. In TASM terminology, these triggers are "planned environments" and "health conditions". Together, they form a "state matrix".

The state matrix comprises individual cells, each representing a distinct state with its workload priorities and management rules, such as filters and throttles. By dynamically transitioning between states in response to real-time system conditions, TASM can better allocate resources and ensure optimal performance, even in fluctuating workloads or resource demands.

In summary, the key difference between TIWM and TASM lies in their approach to workload management. While TIWM relies on static workload definitions, TASM dynamically adjusts workload management based on system conditions (time and events). This allows TASM to provide a more flexible, responsive, and efficient approach to managing workloads in a Teradata environment.

Additionally, advanced **utility throttling** in TASM helps control the number of concurrent load and export utilities running on the system, preventing resource contention and ensuring optimal query performance.

In summary, while TIWM offers basic workload management capabilities for smaller-scale environments, TASM provides advanced functionality suitable for larger, more complex Teradata Database environments with greater workload management requirements.

5.4 Common Characteristics of TIWM and TASM

Classification Criteria

TIWM and TASM use classification criteria to map incoming queries to one of the existing workloads based on their technical characteristics and additional information such as user ID and login account. Query characteristics can be grouped into three main categories:

- The Source of the Query: User, Account, Application, ClientID, Profile, Queryband, and IP Address.
- The Target of the Query: Database, Table, View, Function, and Stored Procedure.
- The Technical Characteristics of the Query: Join Type, Estimated runtime, Statement Type, Estimated Final Row Count, Percentage of rows accessed, Load Utility Type (Fastload, Multiload, etc.), and Number of AMPs involved.

Filters

Filters can be employed to prevent inappropriate queries from the execution or to issue warnings that help identify problematic queries. For instance, a filter can be set up to block reporting user logins between midnight and 6 am, leading to an error message if any reporting user query is attempted during this time frame.

Another example is using a filter to log a warning in the DBC.DBQLOGTBL table when a request consumes more than 100 CPU seconds. This filter allows administrators to monitor queries requiring significant processing power, helping them identify potential performance bottlenecks and optimize resource allocation for better system performance.

Throttles

In Teradata Workload Management, throttles are vital in concurrency control and performance optimization. Throttles limit the number of concurrent sessions, requests, or utilities to protect critical resources such as memory, AMP worker tasks, and CPU from being exhausted. This article delves into the different types of throttles, their applications, and how they contribute to better performance and resource management in Teradata systems.

Throttling Requests

Request throttling limits the number of concurrent requests or queries, ensuring that the defined request limit is maintained at any given time. When the request limit is reached and additional queries are ready for execution, they are either added to the delay queue or rejected immediately. Workload management uses a simple counter to track the number of executing requests. As the request counter falls below the limit, the first query waiting in the queue is executed.

Request throttles can be defined at the workload or system level. Workload throttles enforce limits for requests executing on behalf of the workload and are applied after requests have been classified into the workload. On the other hand, system throttles enforce limits on the system level, applying to every query without further request classification.

Enforcing Session Limits

Session limits restrict the number of parallel sessions allowed at one time. Unlike request throttles, sessions cannot be delayed. No more sessions can be logged on when the session limit is reached and logins are rejected. Session limits can be applied for the complete system or classification targets, such as user, account, or IP.

How Delay Queues are Designed in Teradata

Teradata has undergone significant changes in managing delay queues to streamline the throttling experience in more recent releases. This article provides an in-depth look at how these changes affect the DelayTime field in DBQL and what it means for interpreting delay times in Teradata moving forward.

Single Delay Queue in Recent Teradata Releases

In previous releases, delay queues were set up independently by type of throttle, with each workload throttle having its dedicated queue. However, more recent releases introduced a single delay queue for all throttles, meaning that queries delayed by system throttles reside in the same queue as queries delayed by workload throttles.

While this change streamlines the throttling process, it also requires reevaluating the DelayTime field in DBQL, which now has a slightly different interpretation than in earlier releases.

WDDelayTime, representing the time a query was delayed due to a workload throttle, remains unchanged in DBQL delay reporting for these newer releases. This field is not found in the

DBC.QryLog view can be accessed using the QryLogTDWM view and other TDWM fields, such as WDID.

In the more recent Teradata releases, DelayTime is the total time a query spends in the delay queue, regardless of whether it is due to one or more system throttles, a workload throttle, or a combination of both. The reason for the delay or the number of throttles contributing to the delay is not considered; only the total wall clock time is recorded.

However, subtracting WDDelayTime from DelayTime to determine when a system throttle delays are no longer possible. In earlier releases, workload and system throttles had separate delay queues, allowing this calculation. However, with the introduction of a single delay queue, queries can be delayed by both types of throttles simultaneously, resulting in an overlap of delay times.

Handling Delays by Both Types of Throttles

When a query is only delayed by a workload throttle, DelayTime, and WDDelayTime metrics will be the same. If a system throttle only delays a query, WDDelayTime will be null, and the time reported in DelayTime is solely due to system throttles.

However, if both types of throttles delay a query, it is impossible to determine each throttle's contribution to the delay accurately. For instance, if DelayTime indicates an hour and WDDelayTime shows 20 minutes, the system throttle may have mandated a delay between 40 minutes and 1 hour, as both throttles could be keeping the query in the delay queue simultaneously.

A question often raised by users is why the DelayTime and WDDelayTime columns in Dbqlogtbl allow null values instead of zero for queries that are not delayed. The reason behind this design choice lies in the logging efficiency of DBQL tables. These tables contain many fields, and the priority has always been to ensure efficient data logging. As a result, the philosophy has been to avoid logging a field if unnecessary.

Most fields in the DBQLogTbl are initialized to zero at the beginning of a request. The code responsible for logging values checks if there are any values greater than zero in a field before logging the data. Consequently, fields with zero values do not get logged and report NULLs. DelayTime and WDDelayTime columns fall into this category.

Difference between Zero and NULL DelayTime

When looking at the DelayTime values in Teradata, it's essential to understand what each value represents.

A NULL DelayTime indicates that the request has not been delayed. In other words, the query faced no delays due to workload or system throttles.

A zero DelayTime value represents a delay greater than zero but less than one second. Teradata used to report DelayTime in hundredths of a second in the past. However, DelayTime is reported at one-second intervals in the current releases, as it has an integer data type.

The reason behind Zero DelayTime

The Teradata system calculates a DelayTime value if it finds that hundredths of a second are greater than zero, as initial values are stored internally in hundredths of a second. Any value from 0.01 to 0.99 seconds will be zero with integer division.

This means that when you see a zero DelayTime value, it indicates a delay, but it was so minimal (less than one second) that it was rounded down to zero when converted to an integer value.

Throttling Best Practices

Dealing with workload throttles is generally easier than with system throttles, as the impact of workload throttles is limited to a group of queries. Workload throttles allow for defining limits for each group of requests, making it simple to identify the impact of these limits. System throttles, however, are more complicated to implement from a classification perspective.

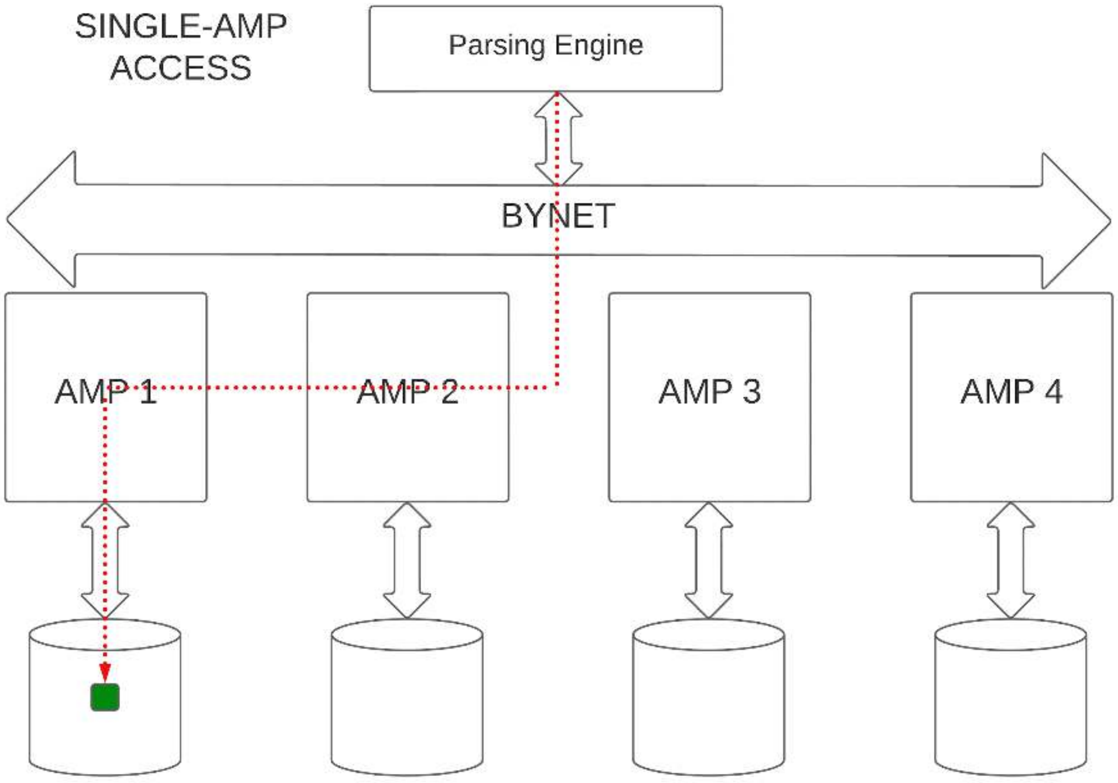
System throttles are best suited for managing situations unrelated to specific workloads but rather the overall system condition, such as periods of system recovery or critical backup and restore activities. In any case, **never apply throttling to tactical workloads**.

Throttles in Teradata Workload Management are essential for concurrency control and performance optimization. By limiting the number of concurrent sessions, requests, and utilities, throttles help protect critical resources and reduce resource contention, ultimately improving system performance. Understanding the different types of throttles, their applications, and how to identify them in the query log is crucial for managing and optimizing Teradata systems effectively.

Common Workload Priorities

Tactical Tier:

This tier has access to all system resources and should be used cautiously, typically for single AMP index accesses. An exception handler automatically assigns any resource-intensive query to a lower priority to protect the system from long-running queries mistakenly assigned to the tactical tier.



Teradata Tactical Workload Exceptions are essential to protect against tactical queries that consume excessive resources. It's necessary to have this protection because the super-priority and almost unlimited access to resources given to work running in the Tactical tier with SLES 11 are easy to abuse.

A Tactical workload exception consists of two different exceptions bundled into one: one exception is on CPU, and the other is on I/O usage. When either of these exceptions is detected, the query running in the Tactical tier will be demoted to a non-tactical workload. Each exception has two threshold variants: a per-node threshold and a sum over all node thresholds.

Each node has its instance of the operating system and the priority scheduler, which only knows about the activities on that one node. Each node's priority scheduler instance monitors its per-node exception thresholds for CPU and I/O. The sum of all node exceptions is monitored and detected by workload management (TASM or TIWM).

To see if a tactical exception has occurred across all nodes, look at the FinalWDID field in the DBQlogTbl. The easiest way to monitor the frequency of tactical exceptions is to check the TacticalCPUException and TacticalIOException fields in the DBQLogTbl.

It is important to monitor the frequency of tactical exceptions and cautiously change the exception thresholds. It's recommended that thresholds for tactical events remain low at the default settings or below.

Tactical queries are expedited, which means they will use reserved AMP worker tasks if such reserves have been defined on the platform. Large numbers of demotions out of tactical could contribute to the misuse of reserved AMP worker tasks and could deplete their number sooner than expected. For that reason, if tactical queries are hitting the exception thresholds often, that should be a tipoff that either the tactical application requires tuning or that the workload belongs in a non-tactical tier.

In conclusion, Teradata Tactical Workload Exceptions are crucial for protecting against excessive resource consumption by tactical queries. It is important to monitor the frequency of tactical exceptions, change the exception thresholds cautiously, and ensure that the tactical queries do not deplete the reserved AMP worker tasks. By following these guidelines, you can ensure that your system runs efficiently and smoothly.

Timeshare Tier:

This tier consists of fixed priorities - Top, High, Medium, and Low. The Top priority receives eight times more resources than the Low priority, four times more resources than the Medium priority, and twice as many resources as the High priority. However, the Timeshare Tier only receives the resources not consumed by the Tactical Tier.

Teradata Tactical and Timeshare Tiers together are powerful tools for optimizing system performance and managing resources in a Teradata Database environment. Administrators can effectively manage workloads by understanding the classification criteria, filters, throttles, and workload priorities, ensuring optimal performance and resource allocation for various tasks and user requirements.

5.6 Teradata TASM Enhancements

Teradata Active System Management (TASM) is an advanced workload management solution for large and complex Teradata Database environments. TASM encompasses all Teradata Integrated Workload Management (TIWM) functionality and adds dynamic workload management features that adapt to changing system conditions in real-time.

This flexibility sets TASM apart from other workload management solutions, such as Snowflake, which primarily relies on scaling rather than these advanced workload management features. This article explores the key components of TASM, including state matrix, state switching, and leveraging the QueryLog for performance analysis.

Dynamic Workload Management Features

TASM distinguishes between two events that drive dynamic changes: time-based events (planned environments) and system-specific events (health conditions).

Planned Environments: Time-based events are scheduled according to defined time windows, activating specific workload management rules when a particular time is reached.

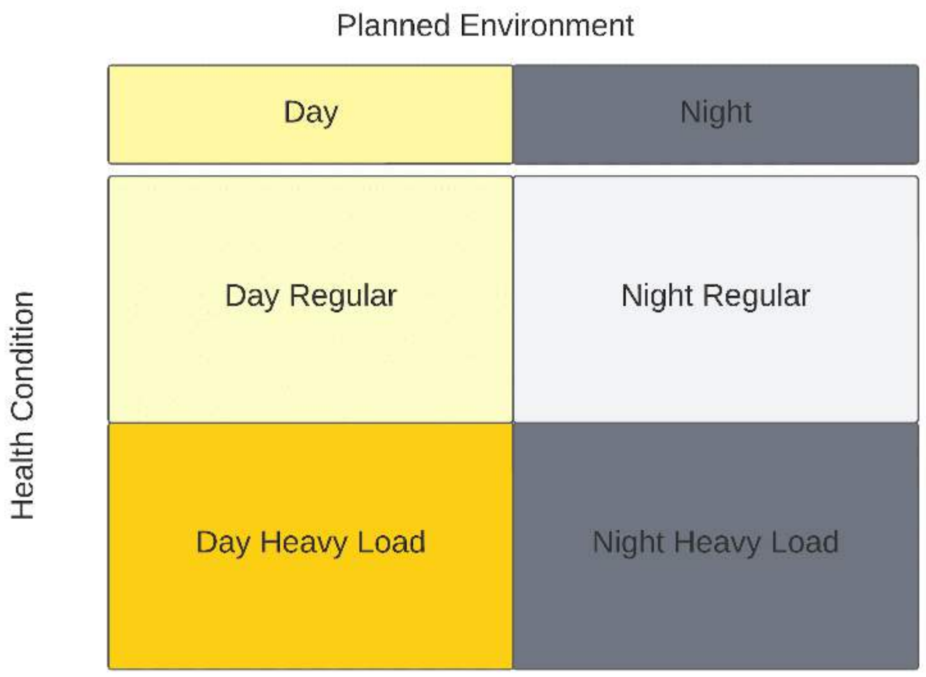
Health Conditions: System-specific events depend on system load and are triggered when certain conditions, such as a specified number of AMP worker tasks (AWTs) in use, are surpassed. Health conditions are not planned, and their occurrence is unpredictable.

State Matrix

TASM combines planned environments and health conditions in a state matrix. Each combination of a health condition and planned event leads to a specific state, representing a set of workload management rules, such as throttles and priorities.

Workload designers can define multiple planned environments and health conditions. The simplest case involves one planned environment (the default "Always") and one health condition

("Normal"). A common approach includes at least two planned environments, one for daytime (protecting reporting users and applications) and another for prioritizing nightly batch loads.



TASM State Switching

TASM regularly checks for relevant events, first assessing health conditions and subsequently evaluating switches to other planned environments. If at least one event occurs, Teradata uses the workload management rules of the related state.

In the case of multiple matching planned environments, Teradata uses the one furthest to the right in the state matrix. If multiple health conditions match, the one nearest to the bottom of the state matrix is used.

TASM Flex Throttles

This is another feature that is only available in TASM.

Typically, throttle limits are set based on available resources and what else is running. However, there may be times when incoming work for the non-throttled activities falls off, and extra CPU is on the platform. In these situations, a flex throttle can be used to release delayed queries so that they can run.

Flex throttles are built on top of one or more already-existing workloads. When active, flex throttles allow delayed requests to be released if resources on the platform fall below a specified usage level. This replaces the need for the administrator to manually release delayed queries or force a change to a different state.

To set up a flex throttle, you need to determine which workloads will participate, decide how many delayed queries will be released each time a “flex event” is reached, choose what constitutes a “flex event”, and select a Flex Throttle Action Interval to tell TASM how long to wait after releasing queries before potentially repeating the action.

Any workload with a workload throttle can be associated with a flex throttle. However, flex throttles are most useful with medium or low-priority workloads. AMP worker task (AWT) availability and CPU utilization trigger a flex event. The administrator controls how many workloads will participate in a flex throttle, what resource levels will trigger the release of queries, and how many queries will be released at a time.

You should start with fewer queries, particularly if the queries are potentially resource-intensive or long-running. The order of the delay queue determines which queries are released when a flex event is triggered. Since the default ordering is first-in-first-out, queries from workloads participating in the flex throttle that has been in the delay queue the longest will be released first.

If you are thinking of trying out flex throttles, start small and use evaluation mode until you have validated that your flex actions are happening at the times you expect and that the number of released queries is appropriate for your environment. Once active, ResUsage can be very helpful in providing information about CPU usage and AWT availability during a flex action.

Flex throttles are a specialized technique to get additional work through the system during under-utilization by releasing queries from the throttle delay queue. As platform resource usage rises, the flex throttle will stop releasing queries, and workload throttles will continue to rely on their workload throttle limits.

The QueryLog and TASM Dynamic Workload Management

The DBC.DBQLOGTBL table provides valuable insights for Teradata tuning experts, making it easy to identify each request's planned environment and health condition under which it was executed. This information can be used to analyze performance issues, identify misassigned queries, and more.

The relevant columns in the DBC.DBQLOGTBL table includes:

- OpEnvId: Planned Environment
- SysConId: Health Condition
- LastStateChange: Timestamp when Teradata activated the planned environment for the request.

Virtual Partitions: While TIWM only offers a single virtual partition, TASM supports multiple virtual partitions, providing greater flexibility in resource allocation.

SLG Workload Tiers: TASM supports SLG (Service Level Goal) workload tiers, which allows for more granular control over workload priorities and resource distribution in TASM.

Teradata TASM brings a new level of sophistication to workload management by dynamically adapting to system conditions through planned environments and health conditions. Utilizing the state matrix and state switching, TASM ensures optimal performance and resource allocation in complex Teradata Database environments. Additionally, the QueryLog offers valuable insights for performance analysis, enabling Teradata tuning experts to identify and resolve issues effectively. It's worth noting that Teradata Workload Management is far more advanced than solutions like Snowflake, which relies almost solely on scaling rather than on advanced workload management features.

5.7 The Teradata Priority Scheduler used in TASM and TIWM

In a previous chapter, we introduced the concept of workload and explored how Teradata workload management classifies requests (queries) into workloads. We briefly mentioned that resource allocation (CPU seconds and IOs) is one of the primary reasons for grouping requests.

We introduced Teradata's two kinds of workload managers (TIWM and TASM) and talked about how they are different and what they have in common.

This chapter will delve deeper into the Teradata priority scheduler and its role in distributing resources across workloads and requests. The priority scheduler is the software responsible for assigning available resources to active requests.

5.7.1 The Teradata Workload Hierarchy

Understanding the Teradata priority scheduler and its role in resource allocation is essential for effectively managing and optimizing Teradata systems.

In Teradata systems, priority levels play a crucial role in managing system resources and ensuring the efficient execution of tasks. The priority hierarchy consists of different levels, each with its unique purpose and characteristics. In this article, we will explore the various levels of the Teradata priority hierarchy and their significance in workload management.

The priority hierarchy in Teradata consists of the following levels:

- TDAT
- User, System, Default
- One or several virtual partitions (TIWM: always one)
- A tactical workload tier (TIWM and TASM)
- Between 0 and n SLG (Service Level Goal) workload tiers (TASM only)
- One timeshare tier (TIWM and TASM)

At the top of the hierarchy, User, Default, and System levels follow TDAT. Teradata system tasks run on Default and System, while user-initiated tasks execute below the User level. Default and System tasks always satisfy their resource requirements before users receive resources, as they are responsible for executing critical system-related tasks.

The User level groups all tasks running on behalf of database users, excluding internal tasks. This separation ensures that critical system tasks receive adequate resources before user tasks, thereby maintaining system stability and performance.

System resources in Teradata Workload Management are allocated from the top of the hierarchy to the bottom. This simple yet effective approach follows the rule that the most critical workloads should be assigned to the top of the hierarchy while the least critical workloads are placed at the bottom.

Below the User level, virtual partitions are used to manage resources among different workloads. Teradata Integrated Workload Management (TIWM) always has one virtual partition. However, in Teradata Active Workload Management (TASM), multiple (up to 10) virtual partitions can be defined.

Exploring Teradata Virtual Partitions

Teradata Virtual Partitions provide an efficient way to allocate system resources among various business units sharing a common Teradata system. This feature, available on Teradata systems running on SLES 11 or higher, is based on the concept of SLES11 priority hierarchies. In this article, we will delve into the concept of Teradata Virtual Partitions, their benefits, and how they function in workload management.

Virtual Partitions enable splitting the system resources at the highest level, allowing different business units to consume their allocated percentage of system resources independently. This feature is particularly useful in multi-tenant environments where multiple stakeholders share a single Teradata system.

It's important to note that the assigned share is not exclusively reserved for each virtual partition. If a virtual partition does not use all its assigned resources at a given time, other virtual partitions can take advantage of the available resources. This approach promotes efficient resource utilization across the system. Although it's possible to define a limit to prevent one virtual partition from using another's resources, this is generally not recommended, as it may hinder overall system efficiency.

While Teradata Integrated Workload Management (TIWM) does not support system splitting and implements exactly one virtual partition, Teradata Active Workload Management (TASM) allows this feature. To take advantage of Virtual Partitions, you will need to have TASM.

Assigning Workloads to Virtual Partitions

Each virtual partition can have several workloads assigned to it. By allocating workloads to specific virtual partitions, businesses can manage their resources independently, ensuring critical tasks receive the necessary resources while maintaining overall system performance. Each Virtual Partition has its priority hierarchy (Tactical Tier, SLG Tier, Timeshare Tier).

Teradata Virtual Partitions provide a robust solution for efficient resource allocation and management in shared Teradata systems.

Tactical Workload Tier

A tactical workload tier is designed for handling high-priority, short-running tasks. These tasks receive resources before lower-priority workloads to ensure timely execution.

Teradata Tactical Workload Tier is crucial in prioritizing tasks within the Teradata system. It offers several advantages, such as expedited and reserved AMP worker tasks, that help maintain optimal system performance.

The Tactical Workload Tier is located just below the virtual partition level in the Teradata priority hierarchy. Tasks assigned to this tier can consume as many resources as required, except for the 5% reserved for the "remaining" workload.

Tactical workloads can interrupt other tasks and take over the CPU quickly, while the interrupted tasks are resumed afterward.

However, assigning long-running and resource-intensive queries to tactical workloads could cut off workloads defined at lower hierarchy levels from resources. Therefore, assigning only "real" tactical requests, such as single-AMP or group-AMP operations, is essential to the tactical tier.

If recommended design approaches have been followed, most of the resources that flow into the tactical level will flow down to the below level. If you use TASM, the next level down will be SLG Tier 1.   If you are using TIWM, it will be Timeshare.

Teradata has implemented a safety mechanism called Tactical Workload Exceptions to mitigate the risk of monopolizing resources by tactical workloads. This mechanism automatically transfers requests running on the tactical tier to a workload further down the hierarchy when a certain amount of CPU seconds have been consumed (adjustable).

Monitoring Tactical Workload Exceptions

It is crucial to monitor these exceptions regularly to identify requests that consistently cause exceptions. Such requests should be moved to lower workloads in the priority hierarchy to maintain optimal system performance and resource allocation.

The Teradata Tactical Workload Tier offers unique benefits that can significantly improve system performance and efficiency. By understanding its features and implementing best practices, organizations can harness the power of tactical workloads and avoid potential risks associated with resource allocation.

Regular monitoring of Tactical Workload Exceptions ensures that resources are allocated effectively, helping maintain optimal performance across the entire Teradata system.

The Service Level Goal Tier (SLG)

In TASM, SLG workload tiers can be defined, with each tier having specific performance objectives. SLG tiers help manage resources for different workloads and ensure they meet performance goals. Up to five SLG Tiers may be defined, although one, or maybe two, are likely to be adequate for most sites. Multiple workloads may be placed on each SLG Tier.

Teradata SLG tiers are vital in managing resources and maintaining efficient performance across the system. In this article, we will delve into the intricacies of the SLG tier in Teradata Workload Management with TASM and discuss its importance, functioning, and best practices for optimizing its use.

The Role of SLG Tiers in Teradata Workload Management

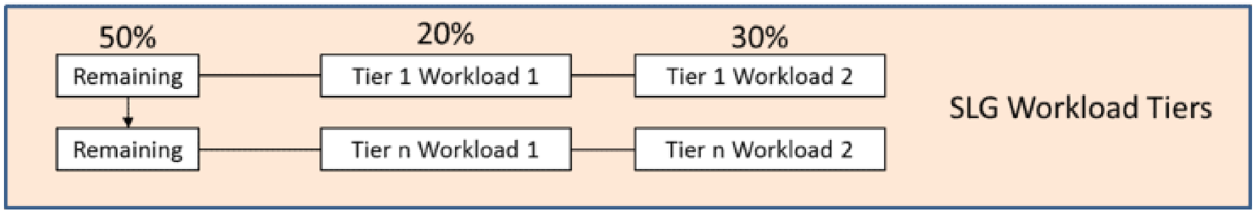
The SLG tiers are positioned below the tactical tier in Teradata Workload Management. Each SLG tier workload is assigned a relative weight, which determines the percentage of resources the workload can consume from the level above in the hierarchy. Additionally, the relative weight defines the top resource limit for the workload. An assigned allocation percent could provide more resources than a workload ever needs.

Unused resources allocated to one workload will be shared by the other user-defined workloads on that tier based on their percentages. What is offered to each is based on the ratio of their workload allocations.

Resource Allocation and Distribution in SLG Tiers

If not all workloads on an SLG tier can utilize their assigned resources, the remaining resources will be equally distributed among other workloads at the same SLG tier in need. Any leftover resources will be allocated to the "remaining" workload and passed down in the hierarchy.

In the below figure, at least 50% of the resources defined in the "remaining" workload will flow down to the SLG tier below tier 1, which could be another SLG level or the timeshare tier, depending on the system setup. If tier 1 cannot use 50% of its assigned resources (20% + 30%), all unused resources will be added to the 50% of the "remaining" workload.



Concurrency and Relative Weights

It is crucial to understand that relative weights allocate resources to workloads. For instance, if "Workload 1, Tier 1" can use up to 20% of the resources received by the previous tier, it does not imply 20% of the entire Teradata system's resources but the percentage of the resources that flow into the tier.

The active requests of a workload share the relative weight, which affects resource allocation for each task. If one query runs in workload 1 of SLG tier 1, it will consume all resources. But if 10 queries run in workload 1 of SLG tier 1, each query will only get 1/10 of the resources.

Concurrency, therefore, has a huge impact in this case. But of course, unused resources from other workloads on tier 1 can be used additionally.

Relative weights are theoretical values representing the allocation of resources a workload would receive under the assumption that no tactical work is active, no work is running outside of the database, and all workloads are active and utilizing their entitled allocations. Such a scenario is improbable; relative weights are only abstract concepts. Nonetheless, they are a useful indicator of how resources will be physically distributed in your environment.

Best Practices for SLG Tiers

Limit the number of SLG tiers: Fewer SLG tiers ensure more stable runtime behavior for requests, as lower hierarchy levels depend on the resources "leftover" from higher levels.

The Teradata SLG Tier in Workload Management is pivotal in managing resources and ensuring system performance. Organizations can optimize resource allocation and maintain a high-performing Teradata system by understanding its functioning and implementing best practices.

Timeshare Tier

The timeshare tier is the lowest level in the priority hierarchy. Tasks within this tier share available resources based on their relative priority. This level is designed to handle workloads that do not require immediate resource allocation or have lower priority than tactical and SLG tiers.

The timeshare tier receives the remaining resources from the lowest SLG tier (or the tactical tier if using TIWM or not defining SLG tiers in TASM). Still, the priority scheduler ensures that a few resources always reach the timeshare tier.

Unlike the SLG tiers, the Timeshare tier implements a distinct resource allocation strategy for workloads defined as timeshare workloads.

The Timeshare tier has four resource allocation weights: Top, High, Medium, and Low. Fixed ratios are defined between these four levels, as opposed to the relative weights used in the SLG tiers:

- Each request in a workload of group "Top" can consume eight times more than any request of a group "Low."
- Each request in the group "High" workload can consume four times more than any request of the group "Low."
- Each request in the group "Medium" workload can consume two times more than any request of the group "Low."

This rule is valid across all workloads defined on the Timeshare tier. While all requests of one workload on an SLG tier compete for the relative resources assigned to that workload, all timeshare requests of all defined workloads compete for the total resources entering the Timeshare tier.

If 5 queries run in Top, each will get 8 times the resource of a single query in Low. If there are 50 queries in Top, each will get 8 times the resource of a single query in Low. As each query in Timeshare Top will get 8 times the resource of any query running in Timeshare Low, it doesn't matter what the concurrency levels are: No matter how many queries are in Top Level, they will always run more efficiently than the queries in Low, and therefore it is recommended that you place workloads in Timeshare based on priority.

Here is an example of how resources are split in the timeshare tier for 4 High and 2 Low queries. The High Queries will always get more resources than the lower level queries:

4*4 + 2*1 = 18

Each High Query: 4/18 = 22,22%
Each Low Query: 1/18 = 5,55%

Now we have 20 queries in the Top and 1 in Low, showing that even with a lot of High Queries still, Low Queries get fewer resources:

20*4 + 1*1 = 81

Each High Query: 4/81 = 4,93%
Each Low Query: 1/81 = 1,23%

There are two possibilities as soon as the remaining resources reach the Timeshare tier:

If the workload defined as timeshare consumes all remaining resources, 100% of the system resources are consumed and distributed.

If the timeshare workloads do not consume all remaining resources, they will be assigned to other workloads in need.

The backflow of resources to the lowest SLG tier from the timeshare tier will be offered to all active workloads on the tier, proportional to their allocations. However, this situation would only occur if Timeshare workloads could not consume the resources that flowed down to them.

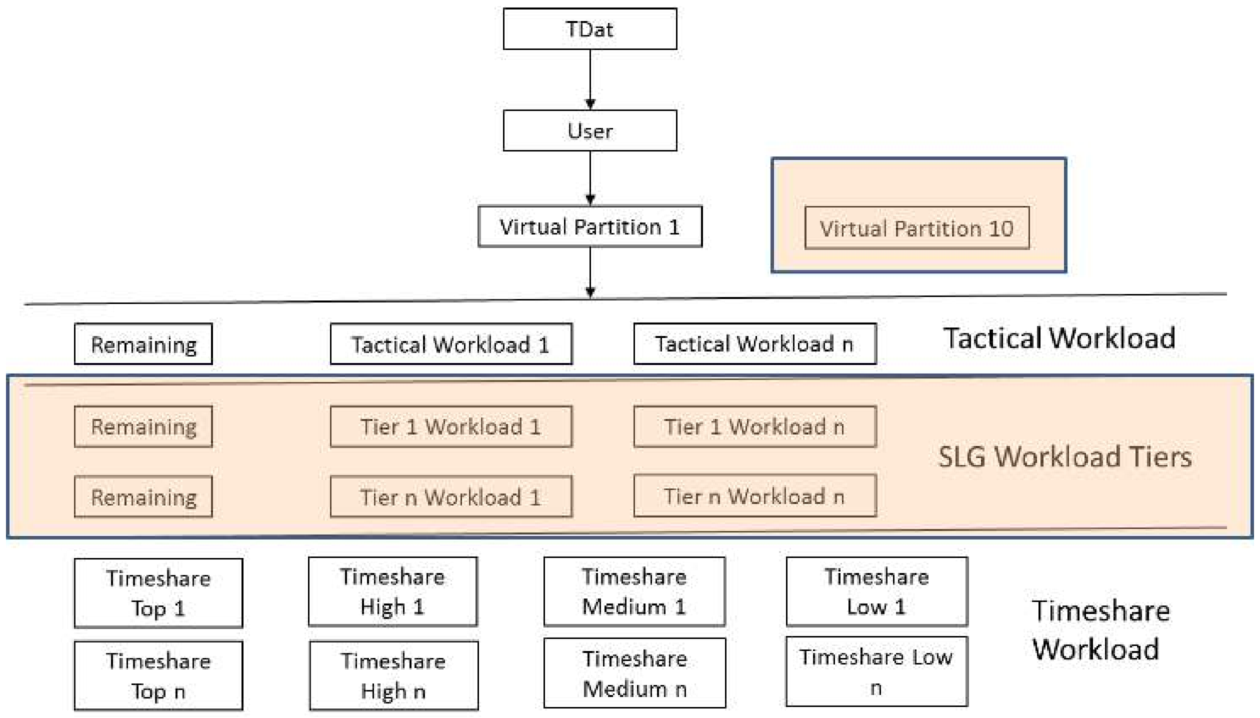
All resources flow down to the base of the hierarchy first. Only if they cannot be used by the workloads at the base will they be available to other workloads to consume. **Every resource is well-spent as long as someone can use it.**

Understanding the Teradata priority levels and their role in workload management is essential for efficient resource allocation and task execution. By organizing tasks within the priority hierarchy, Teradata systems can ensure that critical tasks receive resources first and that user tasks are efficiently managed according to their priority and performance goals.

Differences in Priority Scheduler Hierarchies between TASM and TIWM

The figure below illustrates the differences between TASM and TIWM priority scheduler hierarchies, with the features only available in TASM highlighted in orange.

Understanding the differences between TASM and TIWM priority scheduler hierarchies is essential for making informed decisions about workload management in Teradata systems. TASM offers additional features, such as support for multiple virtual partitions and SLG workload tiers, making it a more advanced and flexible option for managing workloads and resource allocation.



Best Practices for Resource Allocation

In Teradata workload management, best practices for resource allocation involve strategically assigning workloads to various tiers and access levels. For tactical workloads, prioritize highly-tuned queries with low service level expectations and avoid assigning load utilities.

Demote non-tactical queries when necessary and ensure adequate resource allocation for tactical workloads. For SLG tiers, assign high-priority workloads with response time expectations but avoid overloading SLG Tier 1 with non-tactical queries. Consider sibling sharing of unused resources for workloads with differing peak time windows.

For Timeshare workloads, allocate most platform resources to the appropriate access levels and prioritize high-priority non-tactical workloads in the Timeshare Top access level if SLG tiers are not used. Place low-priority workloads in Timeshare Low and maintain minimal concurrency.

Consider that the same query might perform better in the timeshare tier than in the SLG tier one level above. Here is an example: You assign 20% to the only workload in the SLG tier, which means 80% of the resources flow into the timeshare tier. Assuming no other workload is active, the query will perform 4 times faster in the timeshare tier than in the SLG tier. However, it is being executed lower in the hierarchy. To avoid such situations, you can classify most queries into the timeshare tier (increasing its concurrency level).

5.7.3 Optimizing your Utility Workload

In our previous articles, we discussed Teradata workload management, focusing on SQL request classification, throttles, dynamic workload management, and the priority scheduler. This article delves into utility workload management in Teradata, which handles non-SQL protocol tasks. We will cover the types of utility tasks, protocols, and techniques used to manage resources and concurrency.

Understanding Teradata Utility Workload

Utility jobs are user requests that do not implement the SQL protocol. Currently, there are four primary utility tasks:

- Loads (Fastload, Multiload)
- Unloads (FastExport)
- Backup & Restore (ARCMAIN, BAR)

These tasks implement their own specialized and optimized protocols, and Teradata supports the following protocols for utility workload management:

- The FastLoad Protocol
- The MultiLoad Protocol
- The FastExport Protocol
- The Backup & Restore Protocol

Several implementations exist for each utility protocol (e.g., FastLoad protocol is implemented by the Fastload Utility, the TPT Load operator, and the JDBC Fastload). Non-Teradata utilities (3rd party) can apply these protocols via TPT, but some workload management features may be unavailable.

Managing Utility Resource Usage

Concurrency Levels: Defines the maximum number of utilities that can be executed simultaneously. If the limit is exceeded, arriving utility jobs will be delayed or rejected.

Utility Session Limit: Defines the minimum and maximum number of sessions each utility job uses.

Concurrency Management in Teradata Utility Workload

Concurrency limits are implemented using throttles, which can be defined at the system and workload levels. Teradata workload management also implements hard limits that user-defined utility or workload throttles cannot exceed:

- Limit for Fastloads + Multiloads + FastExports: 60
- Limit for Fastloads + Multiloads: 30
- Limit for Fastloads: 30
- Limit for Multiloads: 30
- Limit for Fastexports: 60
- Limit for Backup & Restore: 350

Additionally, Teradata has a limit for AMP worker tasks (AWTs), allowing utilities to use only 60% of the available AWTs at one time. AWT limits may cause utility jobs to be delayed or rejected, even if concurrency limits are not reached.

User-defined utility throttles can manage resources at the system level, and utility workloads using their throttles can be defined for more detailed workload management by adding "utility" as a classification criterion, along with other measures like "request source," "target," and "query band."

Session Management in Teradata Utility Workload

Limiting the number of utility sessions is another way to manage system resources. Teradata workload management automatically handles sessions unless user-defined session limits overwrite them. Workload management distinguishes between small, medium, and large data amounts, with medium-sized data being the default expectation.

Default session limits can be adjusted using the query band, and user workloads can be defined with specific classification criteria, such as Utility Name, Request Source, Query Band, and Data Size (via the "UtilityDataSize" query band).

Teradata utility workload management is crucial for managing non-SQL protocol tasks and ensuring optimal system performance. By understanding the types of utility tasks, protocols, and resource management techniques, users can design efficient utility rules and workloads to address unique situations and requirements.

Understanding resource allocation rules and best practices in Teradata Workload Management is crucial for optimizing system performance and effectively managing workloads in a Teradata system. By following the hierarchical structure and implementing the safety mechanism, you can ensure that resources are efficiently allocated to all workloads according to their priority.

About the Author

Roland Wenzlofsky is a seasoned data warehousing and analytics professional with extensive experience in the industry. As the driving force behind www.dwhpro.com, he has established a platform dedicated to sharing knowledge and best practices in data warehousing, Teradata systems, and related technologies. Through his website, Roland provides valuable insights, tips, and resources for database administrators, developers, and data professionals seeking to enhance their skills and stay updated in the ever-evolving field of data management.

With a deep understanding of Teradata systems and various data warehousing concepts, Roland has contributed significantly to the growth and development of the data warehousing community. His passion for sharing knowledge and helping others succeed in their data-driven endeavors has earned him the respect and appreciation of his peers and loyal readers. As the author of numerous blog posts, articles, and educational materials, Roland Wenzlofsky remains a trusted and influential voice in the world of data warehousing and analytics.