

# How can clients **Master** their complex z/OS **workloads?**

By Daniel Rosa and Donald Schmidt

Last Updated: 04/03/20

With the high utilization shared mainframe environment, it is critical to know how the many interdependent workload components are interacting with one another over time. Work arrival pattern changes can negatively impact workload component availability. IBM z/OS Workload Interaction Navigator (abbreviated as Navigator) specializes in dynamically recognizing the full impact of work arrival pattern changes over time.

Identifying and correlating transient activity changes in run-time execution - *across the z/OS stack* - enables analysts to **learn** key insights into **why** the problem occurred.

Both CICS and IMS have issued a Statement of Direction to provide their transaction level, run-time execution activities.

## ***Imagine if a client could quickly identify which impactful workload components were performing anomalously.***

A major inhibitor to gaining this insight is that data is only generated at the two extreme ends of the spectrum - Too coarse RMF 15 minute interval data versus too fine-grained Middleware per transaction data.

A second major inhibitor is the lack of standardization. Each Middleware has its own unique form of fine-grained data generation, and there is no easy way to correlate activities across the often interdependent Middleware workload components. It is difficult to identify the root cause of complex problems when you can only see one puzzle piece at a time.

## ***Navigator technology overcomes both of these inhibitors, by generating:***

- 1) Summarized data at a high frequency
- 2) in a Standardized format across the z/OS and Middleware stack.



Navigator Icon for Zowe

Traditional Middleware monitor technology leaves the decision on how to summarize the data, as an exercise for the problem analyst. That design incurs high overheads both in CPU and disk costs to not only produce the detailed data, but also to reduce that data into a human consumable summarized format. Often the analyst doesn't know how to best summarize the data to yield the most valuable insights. And then because each monitor is Middleware specific, there is no way to directly compare the individually summarized data.

Navigator technology takes a radically different approach in that it ***prescribes how the data is summarized for analysis, as part of its data generation.*** This enables low cost, context rich data generation, while requiring only a lightweight ***Visually Intuitive analytics engine.***

***One goal of Navigator technology is to move z/OS Performance analysis from the difficult Art it is today, toward a cost-effective Science.***

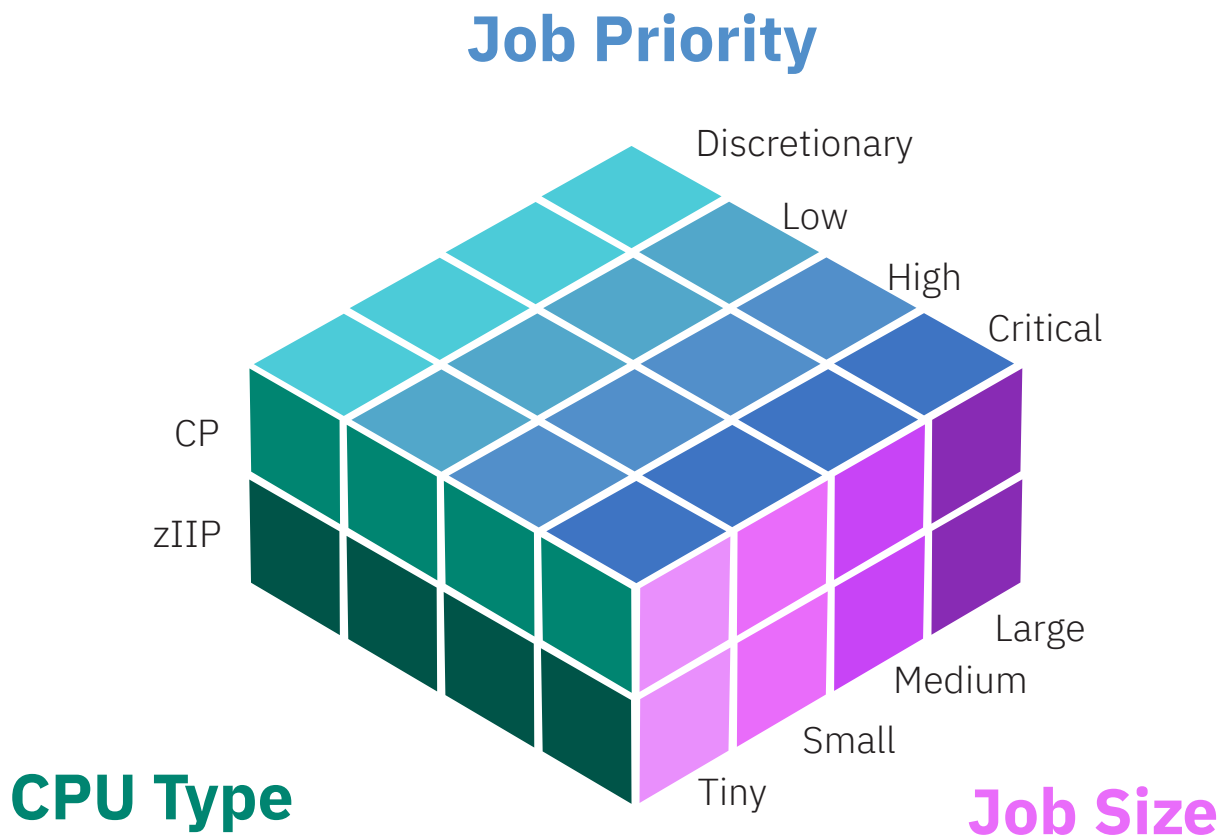
(both in data generation, and analysis costs)

# Smarter Data Generation

IBM z/OS Workload Interaction Correlator (abbreviated as Correlator) **first** defines a standardized way to efficiently summarize key performance indicators, at a high frequency - every 5 seconds. The high frequency summary data enables Interacting Activity baselines to be established for what is normal for a workload component. This enables transient periods of anomalous activity to be identified, from those baselines. These transient Interacting Activity deviations (i.e. Peaks / Valleys) from their baselines are called Interacting Trends.

Standardizing how data is summarized includes defining a methodology to reduce the many workload complexities, down to a smallish set of interacting workload components. A workload component consists of multiple z/OS jobs that share a like Job Priority, and Job Size. There are four Job Priorities: Critical, High, Low, and Discretionary, and four Job Sizes: Large, Medium, Small, and Tiny, for up to 16 unique workload components. Every 5 seconds all running jobs are classified with both a priority, and a size.

# The Correlator transforms workloads from a black box into *Workload Components*



## ■ Generates Standardized, Synchronized, Smarter Data - Every 5 seconds

- ▶ Provides holistic, compartmental, exceptional activities with portions
- ▶ Summarizes key activities by workload component (cube) with exceptional job activity and job name per workload component activity

## ■ Enables Navigator analytics across the z/OS software stack to:

- ▶ Dynamically identify and correlate transient anomalous activities
- ▶ **Implicate or exonerate Workload Components**
- ▶ **Correlate impactful Cause and Victim peers**

## ***There are two levels of data aggregation:***

- 1) At the job level where activities (e.g. number of transactions, response times) are aggregated;
- 2) At the workload component level where activities for all jobs with a given classification (e.g. *CP, Critical priority, Large size*) are aggregated.

Once aggregated per workload component at the 5 second level, the data is summarized (e.g. average response time / transaction). A single context rich SMF record is then written that includes the summarized data for all workload components found during that 5 second period. That record may summarize millions of transaction for that period. All participating z/OS components, and Middleware use this standardized workload component classification, to produce their own SMF record containing their tracked activities.

Specific workload components are identified by three attributes -- CPU type, Job Priority, Job Size (e.g. CP, Critical, Large). A Job Size of 'All' (e.g. CP, Critical, All) represents a collection of workload components.

The final key piece of data provided in the SMF record, is the worst offending job for a workload component Activity. Remember there are two levels of data aggregation, with the first being at the job level. At the end of each 5 second period, the job within each workload component that had the maximum value for an Interacting Activity is included in the record.

The **second** form of standardization requires synchronizing the SMF records from all the participating z/OS and Middleware. At the top of each minute and every 5 seconds thereafter, a single trigger drives writing SMF records for all participating peers. This record synchronization enables activities from across the z/OS stack, to be correlated in ways that were never before possible.

These strongly structured data generation techniques **enable impactful anomalous activities to be temporally correlated, across the entire z/OS and Middleware stack**. Understanding correlation is a key step toward generating problem insights.

Collecting this low CPU cost data generation provides insight generating, always on *First Failure Data Capture*.

# Visually Intuitive Analytics Engine: IBM z/OS Workload Interaction Navigator

The insights gained from this Visually Intuitive Analytics Engine enables a mind set change in that when a client is armed with the right information, they can not only reduce their problem triage time-to-action, but also prevent future outages.

As noted above each Correlator participant produces their own SMF record. Either an individual participant or all available records can be analyzed, by specifying the Analysis Scope.

Never before has so much context rich z/OS and Middleware data been available on a single insight generating tool. With great power, comes great responsibility. A key design point for this tool is that it's automated analysis focuses the analyst on only the most impactful issues. This automated Navigator technology uses client specific data, to identify client specific anomalies.

Because of the patented techniques used to generate the data, only a lightweight browser Analytics Engine is required to generate actionable insights.

Armed with the correlation capability described above, the Navigator can quickly identify **‘Cause and Victim’** peer workload components, using automated analysis techniques. For a 15 minute interval, impactful Interacting Activities that have transient deviations from their baselines, are dynamically identified. Activities with non-deviating baselines are excluded. Then Activities with (reoccurring) Peaks / Valleys over time are temporally correlated to other Activities, that have matching Peaks / Valleys. The workload components associated with each of the correlated Activities are identified as ‘Cause and Victim’ peers. The correlated Activities themselves describe how those peers interacted with one another. This intuitively describes the full impact of run-time execution changes that occur over time.

For example, reoccurring peaks in CPU time for a workload component that includes DB2, may be correlated to a CICS or IMS workload component’s reoccurring CPU time valleys. With the push of an automated button - the DB2 and CICS or IMS workload components are identified as ‘Cause and Victim’ peers.

**A second form of correlation** that Navigator technology enables is mapping System Resource use to Consumers. An operating system provides services to perform common functions. Often, these services obtain a System Resource like a lock to serialize ‘critical path’ updates. Jobs invoking these services become Consumers of System Resources. Correlator participants that generate both individual System Resource and Consumer Resource Activities, enable the Navigator to correlate anomalous peaks between them, to show which Consumers are driving System Resource use.

The standardized data collection **also enables the Navigator to do direct comparisons across different intervals in time**, to identify workload component interaction differences between those intervals. One example is if a problematic interval is detected at 9:00 AM on Monday, then the same interval from the previous Monday could be used as the Training interval, for the problematic interval. This Training Interval technology goes a step further than the single interval technology described up until now.

Specifically, Interacting Activities that would have been identified as anomalous in the training interval, if single interval analysis was performed against it, would now be treated as ‘normal’ when using that interval to train the Navigator for a problematic interval.

# The Navigator enables clients to quickly implicate or exonerate workload components - *with the push of a button.*

The Navigator's Proactive Change Verification enables clients to verify - *any change* - had the desired effect. To do this Correlator data is captured both before and after the change, then the Navigator's two interval analysis is used to determine the run-time execution differences between the two (if any). This can be used to validate software maintenance (e.g. Recommended Software Upgrades), software upgrades, and hardware migrations. This technology has a higher degree of precision, when a 'repeatable' workload with significant CPU demand is run both before and after the change. **Proactive Change Verification allows early problem definition - before causing an outage.**

As noted above, **arrival pattern changes** may negatively impact workload component availability. The configuration may unintentionally set latent traps that are not triggered during the normal workload. A workload change (e.g. increase in arrival pattern) can trigger latent traps to spring, and cause elongated response times and / or workload component outages.

While clients can't control work arrival patterns, armed with the insights from this tool they can **manage the consumption rate of work** within their system. Too much concurrency causes system-wide traffic jams, which can result in unpredictable elongated transaction response times, as seen in problematic RMF intervals. Targeted reconfiguration of a workload component to 'governor' its throughput has always been possible, but until now there was no data available to enable an informed decision. The Navigator provides that key feedback.

Informed governor configuration may lower a client's software licensing costs, by avoiding unnecessary peaks in CPU utilization.

Clients routinely taking the proactive steps described above, at a minimum better prepare their analysts on how to gain valuable insights from this powerful analysis tool, for the (hopefully) rare outage - when a timely response is required.

With these holistic Navigator insights, **clients are able to make informed decisions that were never before possible, to yield higher availability and more consistent response times** - thereby effectively improving their workload volatility resilience.

This lightweight, easy to use tool provides the opportunity to make every functional layer more informed (or smarter), from code development and test, to workload component configuration, to performance problem detailed analysis.



# Using the Workload Interaction Navigator

## Configuring Navigator Analysis

The Navigator performs single interval or two interval analysis and requires input Correlator interval(s) be 60 minutes or less and from a single system. The top of the Navigator contains configuration information about the interval(s). Hovering the cursor over the graph, buttons, and other text often provides help text to aid an analyst. For more detailed help information, search this whitepaper for keywords of interest.

In single interval analysis, the Navigator shows a progression of anomalous Interacting Activities. In two interval analysis, the Navigator shows significant differences between anomalous and base intervals. After loading the input Correlator interval(s), the Navigator's analysis screen presents a graph with time on the X axis and a stacked CPU utilization graph by job priority on the left-Y axis. Hovering the mouse over the graph shows the average CPU utilization for each 5 second period. For two interval analysis, the graph defaults to the anomalous interval as shown by the toggle button at the top left of the configuration information. At any point, an analyst may use the toggle button to view current content for the other (e.g. base) interval.

For each input Correlator interval longer than 15 minutes, the analyst must choose an analysis interval page (e.g. 1, 2, 3, 4). For the remainder of this section, analysis interval means analysis interval page. For two interval analysis, choose an analysis interval for anomalous and base intervals using the toggle button, and begin analysis in the anomalous interval.

The Navigator defaults to an Analysis Scope of All to analyze all Activities from all z/OS components and middleware products generating data. This allows the Navigator's Automated Analysis to identify, correlate, and prioritize anomalous Activities across the z/OS software stack. To limit activities to a specific z/OS component or middleware product, set Analysis Scope.

The Navigator provides a Workspace for Beginner and Experienced analysts. The Navigator defaults to Workspace Experienced which provides access to all data and all forms of analysis. With Workspace Experienced, an analyst may manually navigate any data available. Workspace Beginner analyzes important basic data using only the Navigator's Automated Analysis. Beginner data is limited to showing the 'Big Picture', most powerful Activities. Workspace Beginner is intended for single interval analysis, to get analysts familiar with the Navigator's Automated Analysis. The z/OS component or middleware product generating data determines whether each Activity is Beginner or Experienced data.

## Navigator Automated Analysis

Always begin with Navigator Automated Analysis.

Navigator Automated Analysis is designed to present contextualized impactful Interacting Activities, that client SMEs need to understand, to gain key insights into why the problem occurred. For two interval analysis, impactful Activities are presented that had differences across the intervals, to gain key insights into 1) what changes occurred, or 2) if the explicit changes made had the desired effect. In all cases, the identified Activities are contextualized both by correlation to other like Activities, and the Activity's workload significance.

The Navigator analyzes every Workload Component Activity in the selected Analysis Scope and analysis interval(s) to establish a normal base-line (e.g. average). On the graph, the Navigator considers Activity values between the horizontal dashed lines as 'normal' for the analysis interval. The Navigator uses these base-lines to identify if significant transient deviations occurred throughout the interval. Activities are classified into three groups:

1. those that have no significant deviations (non anomalous);
2. those that have significant peaks;
3. those that have significant valleys.

For those Activities with transient deviations, the Navigator calculates an Activity Volatility value that summarizes anomalous peaks or valleys outside the dashed lines. Activity Volatility values are either positive or negative indicating peaks or valleys are more significant. Larger absolute values indicate larger peaks or valleys. The Navigator uses Activity Volatility as a key criteria in determining whether an Activity is anomalous. Activities that are non-anomalous are assigned an Activity Volatility of NaN (Not a Number). Activity Volatility is only comparable for the same Activity in a different interval (e.g. for 2 interval analysis, 'CPU Time' Activity Volatility is comparable between the base and anomalous intervals). Activity Volatility is not comparable across different Activities (e.g. Activity Volatility between 'CPU Time' and 'Input/Output Requests [I/O]' is not comparable).

For single interval analysis, Automated Analysis Type defaults to 'Interacting Trends' where the Navigator temporally correlates anomalous Activities by their peaks and valleys into Correlated Activity groups (aka **Cause and Victim Activity peers**). The Navigator prioritizes Correlated Activity groups by the number of Interacting Activities in each group – the more Interacting Activities, the more impact the group has on the workload. If a reported problem triggered analysis, an analyst may need to review multiple groups to find which group contains Activity anomalies that match the problem signature. Within each Correlated Activity group, the Navigator prioritizes Global Activities (e.g. Activities with no Workload Component or a general Workload Component like CP/All/All) that impact the whole workload ahead of Workload Component Activities (e.g. CP/Critical/Large) that impact a subset of the workload. The Global and Workload Component Activity set within each Correlated Activity group is sorted by Activity Volatility.

For two interval analysis, the Navigator learns what is 'normal' from the base interval, to identify Activities whose averages, or volatility have changed across analysis intervals. The Automated Analysis type defaults to 'Trained - Increasing Deltas', where the Navigator searches for run-time execution differences, to identify those Activities that Increased in significance, or volatility. This default applies most often when the CPU utilization in the anomalous interval increased from the base interval. Two types of groups are required to classify the differences:

1. Delta Average Activity group - significance;
2. Delta Correlated Activity groups - volatility.

The (single) Delta Average Activity group is the first and most impactful, where Activities with Average Delta increases are listed. Similarly to above, Global Activities are followed by Workload Component Activities, with both Activity sets sorted by Activity Average delta.

To form the second group, the Navigator identifies Activities whose Activity Volatility classifies them as:

- Anomalous in the anomalous interval and not anomalous in the base interval.
- Anomalous in the base and anomalous intervals with significant Activity Volatility increase. The Volatility Delta value summarizes the Activity Volatility change between the base and anomalous intervals.
- Anomalous in the base and anomalous intervals with opposite Activity Volatility signs. One interval has positive Activity Volatility indicating peaks are more significant and the other interval has negative Activity Volatility indicating valleys are more significant. For this case, Volatility Delta displays >9999%.

The Navigator then temporally correlates these Activities by anomalous interval peaks and valleys into Delta Correlated Activity groups. Delta Correlated Activity groups are prioritized by the number of Interacting Activities in each group. Similar to above, Global Activities are followed by Workload Component Activities with both Activity sets sorted by Activity Volatility.

At times the analyst may need to change from the default two interval Analysis Type to 'Trained - Decreasing Deltas'. This Analysis Type may be more appropriate when the CPU utilization Decreased within the Anomalous interval. It must be noted that the changes across intervals may NOT be homogeneous, in that there may be both Activity Increases, and Decreases across the intervals. For complex problems, an analyst may need to understand both sets of changes, to gain the full insights available.

To begin Navigator Automated Analysis, click the Analyze button. This action reduces, correlates, and prioritizes Interacting Activities into Activity Groups, loads the first Activity Group, selects the first Interacting Activity, and graphs it on the right-Y axis. Hovering the mouse over the graph shows the Activity value and Activity details such as the most exceptional jobname. Use the Next (>) and Previous (<) Activity buttons to step through each Activity in the Activity Group. Alternatively the analyst may click on a row within the table to graph a specific Interacting Activity. For the last Activity in the Activity Group, the Next (>) Activity button advances to first Activity in the next Activity Group. The Next (>) and Previous (<) Group buttons (on either side of the group name) move to the first Activity in the next or previous Activity Group.

For two interval analysis, actions against the automated analysis Activity table are prohibited when the base interval is current interval, to ensure a base Activity is not interpreted as an anomalous Activity. To continue Automated Analysis, toggle to the anomalous interval.

### **Other Analysis Techniques:**

An analyst may temporarily deviate from Automated Analysis by using one of the following:

#### **Comparing Activities:**

To compare two activities on the same graph, choose one Activity and on the Home tab, click 'Pin Activity' to save the current Activity in the graph. Then select other Activities to visually compare the new and pinned Activities. Clicking 'Pin Activity' again removes the pinned Activity from the graph.

#### **Correlating Activities:**

To correlate an interesting Activity with other Activities, choose one activity and on the Home tab, click 'Pin Activity' to save the current Activity in the graph. Then set 'Correlate With' to one of the following:

- Peaks to correlate Pinned Activity peaks to other Activities' peaks and valleys. The Pinned Activity must have a positive Activity Volatility value to find matches. Activities are prioritized in descending order by the number of pinned peaks correlated.
- Valleys to correlate Pinned Activity valleys to other Activities' peaks and valleys. The Pinned Activity must have a negative Activity Volatility value to find matches. Activities are prioritized in descending order by the number of pinned valleys correlated.
- Traditional to correlate the Pinned Activity to Activities exhibiting a similar pattern. There is no Activity Volatility requirement to find matches. Activities are prioritized in descending correlation order. Correlation values may be positive for direct, or negative for inverse correlation.

#### **Timestamp Analysis:**

To find all peaks and valleys occurring at an interesting 5 second period, click on the interesting point on the graph to add the timestamp to the Timestamp list under the Home Tab. Then change Analysis Type to By Timestamp. The Navigator will identify all Activities with a peak or valley within 5 seconds of the selected Timestamp, group Activities into the Timestamp Highest Peak Activity Group or Timestamp non-Highest Peak Activity Group. Each Activity Group is then sorted by the Activity Volatility.

#### **Manual Analysis:**

With Workspace Experienced, an analyst can manually analyze any Activity in the current analysis interval. In two interval analysis, use the toggle button to make the base or anomalous interval current. To manually analyze an activity, click the Advanced Tab and set Scope, Group, Activity, and Workload Component including CPU Type, Job Priority, and Job Size. The Next and Previous buttons are disabled until a subsequent Automated Analysis action (either 'Analysis Type', or 'Correlate with') is chosen.

## **Reset:**

The Reset button resets the Navigator for Automated Analysis and keeps the custom session settings in effect (if any). Click Analyze to restart with the default Navigator Automated Analysis.

## **Navigator Analysis Controls:**

An analyst can configure the data available for Navigator analysis using the following:

### **Analysis Scope**

To do analysis with activities from 1 specific z/OS component or middleware product, select the Home tab and set Analysis Scope appropriately. Then the Navigator will only analyze data within the specified Analysis Scope.

### **Focus Area**

For an Analysis Type that produces Correlated Activity groups, require the Navigator only present groups with at least 1 Activity from an Analysis Scope, or with a specific (partial) jobname. This is useful to further reduce the Correlated Activity groups, to only those that meet a predefined problem criteria (if any). To use this analysis capability, select the Home tab, set Focus Area, activate focus area (click the circle next to add a checkmark), and click Analyze.

### **Bookmark**

To save an Activity to review later, select the Advanced Tab, click the Bookmark icon, and add a description. Bookmarks are most useful when not using Navigator Automated Analysis. Bookmarks remain until the analysis session for the input Correlator interval(s) ends.

### **Settings**

The Gear icon controls Navigator Automated Analysis data and anomaly thresholds for the current analysis session. When the Gear Icon is white, the defaults are in effect. When the Gear icon is orange, one or more settings have been changed from the default.

## Settings Data Controls:

### **CPU Types** (Default: Include all CPU Types)

Include Activities for all CPU types, only the zIIP CPU type, or only the standard CP CPU type in Navigator Automated Analysis.

### **Customizing Excluded Activities** (Default: Include All Activities)

By default, the Navigator considers all Activities. To exclude Activities, change the select box to 'Exclude All the selected Activities', click the plus (+), and choose the Scope(s) or specific Activities to not consider.

### **Include Additional Job Detail** (Default: Exclude Additional Job Detail)

By default, additional job detail is excluded. To include additional job detail, click the circle to add a checkmark. Additional job detail includes the exceptional job's Address Space Identifier (ASID).

### **Exclude Discretionary Priority Workload Components** (Default: Exclude Discretionary)

By default, Discretionary priority workload components are excluded from Correlated Activity groups. To include them, click the circle to add a checkmark.

## Settings Anomaly Controls:

### **Trained Activity Delta Percent** (Default: 8)

For two interval analysis using an Analysis Type of Trained – Increasing or Decreasing Deltas, the minimum Activity Average Delta percent between base and anomalous intervals required for a non-Global Activity to appear in the Delta Average Activity Group. The minimum Average Delta percent for a Global Activity is half of the specified value. Increasing this value may reduce the number of Activities included in the Average Activity group.

### **Target Activities for ‘Scope All’ Activity Average** (Default: 10 \* [SMF 98 subtypes within interval])

For two interval analysis using an Analysis Type of 'Trained ...', the target number of activities in the Average Activity Group for Scope All. Decreasing this value from its default, may reduce the Activities listed in the Average Activity group to the most impactful ones.

### **Minimum Portion Percent** (Default: 5)

In order for a Correlated Activity group to be included, the members within that group must on average meet the Minimum Portion Percent threshold. The intent is to exclude groups that only have a trivial impact. Increasing this value may reduce the number of included Correlated Activity groups.

### **Target Number of Peers** (Default: 2)

Minimum number of peers required for the Navigator to include a Correlated Activity group. Increasing this value may reduce the number of included Correlated Activity groups.

### **Required Peak / Valley Match percent** (Default: 60)

The percent of peaks and valleys that must match between Activities to be included in a Correlate with Peaks or Valleys group. Decreasing this value may increase the number of found matching Activities.

# Detailed Activity Definitions

The ability to summarize the workload's complexities, is a key principle of this technology. Jobs with like attributes - Priority and Size - are grouped into workload components.

Activities for workload components are graphed, and are used to establish which workload components are interacting with one another. Transient peaks or valleys within a graphed Activity for a Workload Component, may occur for multiple reasons. One cause is that the number of Jobs contributing to that Workload Component, transiently changes over time (i.e. a workload mix change). The number of Jobs Contributing often can be determined by hovering over points within the graph. A second cause is the number of contributing Jobs stays constant, and the work arrival pattern transiently changes for those Jobs. All Activities listed below may be impacted by either of these as transient or long-lived conditions.

There are two classes of Activities:

1. Activities that track a simple metric - that generally represents amount of work done (e.g. CPU Time);
2. Activities that track a *compound* metric - where an average is calculated for each point in time (e.g. average Response Time per Transaction).

While performing interaction analysis, it is helpful to understand the significance of the Workload Component being graphed. The significance of a Workload Component is used by both the Navigator's Automated Analysis, and the analyst, to determine if a Workload Component warrants further analysis. More significant workload components may be chosen over less significant ones, as the significant ones are more likely to be impactful to the overall workload.

The term 'Activity Portion' is used to represent the significance of a Workload Component, from the perspective of one Activity. Cross referencing Activity Portions across Activities for a Workload Component, provides a multi dimensional view of that component's impact. An Activity Portion is calculated by dividing the workload component's contribution to that Activity, by the workload-wide total for that Activity. When calculating an Activity Portion for *compound* Activities, the Activity's contribution is always mapped to one of the simple metrics used to calculate the *compound* metric. Averages inherently do not have portions, but the simple metrics used to calculate averages do.

The Navigator Contracts multiple rows in the table, when the Activities for those row have very similar graphs. One benefit from doing this is that when the rows are expanded, the context from the contracted Activities, provides additional insights into why the Activities changed over time. If the Activities weren't correlated into contracted groups, then it would harder for the analyst to derive that level of insight. A second benefit is that non-expanded Contracted rows, enables a higher concentration of uniquely graphed Activities on the screen.



The 'Hide' column within Automated Analysis generated tables, enables analysts to temporarily hide a row from the table. Analysts use this function when they determine that a row is less relevant to solving the problem, than other rows within the table. This may also be used to hide rows that duplicate the information of other rows within the table. This enables the analyst to reduce the set of Activities to only those that are most valuable. Rows remain hidden until either a subsequent Reset, or the specific Automated Analysis configuration is changed (e.g. a Settings update).

The 'See More' action for a row in the table, enables an analyst to see additional context for the Activity being graphed. The amount of context varies with the attributes of the Activity, and the number of intervals being analyzed. For single interval analysis, context for only that interval is shown. Whereas for two interval analysis context from both intervals is shown, to enable direct comparisons of the differences between the two intervals.

When the Workload Component column for the Activity is non-blank in the row, then an 'Activity significance and context' table is displayed. This table provides context for the current Activity, and workload components that are related it. Each row includes the workload component's Activity Portion, and its Activity Average, to enable the comparison of these values, across rows in the table. This is particularly useful during two interval analysis. As noted in the Supervisor Scope 'CPU Time' Activity (below), the 'Job size' Workload Component attribute for a Job may change over time, due to changes in that Job's CPU consumption. The related Workload Component table enables analysts to determine if changes that occurred between the two intervals, were due to Jobs changing their 'Job size' attribute across the two intervals, or not (i.e. Jobs maintained consistent 'Job size' attributes). Rows that have blank in the Workload Component column, have no need for this table.

# Supervisor Scope

Within the Supervisor Scope a number Activities track Lock contention metrics. At times software requires a Lock to serialize its 'critical path' processing, when at most one task can be running at a time. This insures its updates are performed atomically (i.e. all or nothing). The Operating System provides services that require Locks to perform their specific function. Three factors determine the amount of contention that occurs on a Lock:

1. the number of processors in the OS image;
2. the granularity of the Lock;
3. the frequency an OS service requiring a Lock is invoked.

Increasing the number of processors, may increase Lock contention delays. For a travel analogy to this: when a multi lane road is reduced to a single lane due to construction, there will be greater delays when the road under construction has three lanes, than when it has two lanes. Fewer coarse grained locks incur higher contention, than - a larger number of - finer grained locks. Finally higher frequency OS service invocation, may increase Lock contention. Many Lock contention peaks are caused by an increased number of tasks invoking a specific OS service concurrently. In some cases, the insights provided by Navigator analysis enables targeted client reconfiguration, to reduce the parallelism of the consumers of the Lock (e.g. configure fewer Jobs and / or tasks - while consistently meeting WLM goals).

There are two classes of Supervisor Scope Locks.

1. Spin Locks - where tasks spin until the Lock is free (which means they do not give up the CPU until they get the Lock).
2. Suspend Locks - where tasks Suspend waiting for the Lock to be free (which means they give up the CPU, until they are later resumed when the Lock is free).

# Supervisor Scope Activities

**Average CPU Delay** (Group: Delays) - This compound Activity tracks the average delays that are caused by multiple work units being simultaneously queued for CPU execution, at each point in time.

- Work arrival pattern changes may transiently increase or decrease the number work units that are run in parallel.
- Significant increases in CPU delays may contribute to decreasing processor execution efficiency, or increasing transaction response times.

**CPU Time** (Group: Detail) - This simple Activity tracks the amount of CPU time consumed by a Job or Workload Component, at each point in time. This CPU time is tracked independently for CPs and zIIPs.

- The 'Job size' Workload Component attribute assigned to a Job, is directly correlated to the amount of CP or zIIP CPU Time that Job consumed, for that point in time.
- The size of a Job may change at different points in time, due to changes in its CPU consumption (e.g. a Job may change its Workload Component attribute from 'Small' to 'Tiny', because it used less CPU).

**CPU Time Per Dispatch** (Group: Detail) - The periodic running of work units by the OS on a CPU (e.g. tasks within a Job), is called a dispatch. This compound Activity tracks the average CPU Time Per Dispatch, which is calculated by dividing a work unit's CPU Time, by the number of dispatches that occurred over that same period, at each point in time. This CPU time is tracked independently for CPs and zIIPs.

- When no software changes have been applied, this value may transiently change due to the batching of items assigned to a work unit (e.g. each dispatch has more items to process).
- Another cause for changes in this value, is when software changes have been made that increase or decrease a task's CPU demands.
- Significant (transient) decreases in processor execution efficiency, due to increased work unit parallelism and / or CPU demand, may be another cause for changes in values for this Activity.

**Consumer Spin Contention Time** (Group: Lock Consumer) - (see Lock description above) This simple Activity tracks consumer Spin Lock contention time by Workload Component and worst offending Lock consumer Jobname, at each point in time.

- This Activity is directly associated with the 'Global Spin Contention Time' Activity, which tracks Spin Lock contention time workload-wide and the worst offending Spin Lock name, at each point in time.
- This Activity is used to map the worst offending Spin Lock consuming Job, to the worst offending Spin Lock name - when the values for both of these Activities peak at the same time.

**Consumer Suspend Contention Time** (Group: Lock Consumer) - (see Lock description above)

This simple Activity tracks consumer Suspend Lock contention time for the worst offending, individual Suspend Lock name and consumer Jobname.

- This Activity is conditionally associated with the 'Suspend Contention Time', which tracks Suspend Lock contention time workload-wide and for the worst offending Suspend Lock name, at each point in time.
- This Activity may be used to map the worst offending Lock consuming Job, to the worst offending Suspend Lock name, if the values for both of these Activities peak at the same time.

**Cycles Per Instruction (CPI)** (Group: Details) - CPI is used to define the processor's execution efficiency for a Job or Workload Component. Increasing the execution efficiency (i.e. lowering the CPI) generally lowers the cost to get work done. This compound Activity tracks the workload component's CPI, at each point in time.

- The CPI may transiently change over time, due to increased work unit parallelism and / or CPU demand within the OS image, caused by work arrival pattern increases.
- Broad CPI changes may also occur at the time of CPC hardware upgrades.

**Dispatches (Group: Details)** - The periodic running of work units by the OS on a CPU (e.g. tasks within a Job), is called a dispatch. This simple Activity tracks a workload component's number of dispatches, at each point in time.

**Global Spin Contention Requests** (Group: Locks) - (see Lock description above) This simple Activity tracks Spin Lock requests encountering contention workload-wide and for the worst offending Spin Lock name, at each point in time.

**Global Spin Contention Time** (Group: Locks) - (see Lock description above) This simple Activity tracks Spin Lock contention time workload-wide and for the worst offending Spin Lock name, at each point in time.

- This Activity is directly associated with the 'Consumer Spin Contention Time' Activity, which tracks consumer Spin Lock contention time by Workload Component and worst offending Lock consumer Jobname, at each point in time.
- This Activity is used to map the worst offending Spin Lock name, to the worst offending Spin Lock consuming Job - when the values for both of these Activities peak at the same time.

**Global Spin Contention Time Per Request** (Group: Locks) - (see Lock description above) This compound Activity tracks average spin time per contention request workload-wide and for the worst offending Spin Lock name, at each point in time.

**Input/Output Requests (I/Os)** (Group: Global) - This simple Activity tracks the global number of I/O requests, at each point in time.

**JOBS Contributing** (Group: Detail) - This simple Activity tracks the number of Jobs that contributed to a Workload Component, at each point in time.

- This Activity is included in graph hover text for many Activities.
- Changes in the number of Jobs contributing to a Workload Component, may impact the current Activity's graphed values, at the point of change.

**Suspend Contention Requests** (Group: Locks) - (see Lock description above) This simple Activity tracks Suspend Lock requests encountering contention workload-wide and for the worst offending Suspend Lock name, at each point in time.

**Suspend Contention Time** (Group: Locks) - (see Lock description above) This simple Activity tracks Suspend Lock contention time workload-wide and for the worst offending Suspend Lock name, at each point in time.

- This Activity is conditionally associated with the 'Consumer Suspend Contention Time', tracks consumer Suspend Lock contention time for the worst offending, individual Suspend Lock name and consumer Jobname, at each point in time.
- This Activity may be used to map the worst offending Suspend Lock name, to the worst offending Lock consuming Job, if the values for both of these Activities peak at the same time.

**Suspend Contention Time Per Request** (Group: Locks) - (see Lock description above) This compound Activity tracks average suspend time per contention request workload-wide and for the worst offending Suspend Lock name, at each point in time.

# CICS Scope Activities

Most of the CICS Activities provide robust graph hover text, where Workload Component level, Job level, and CICS transaction name level values are displayed. The Job and transaction name levels represent those with the worst offending (i.e. maximum) value for the graphed Activity.

**Cpu Time** (Group: Detail) - This simple Activity tracks the amount of CPU time consumed by this CICS Workload Component, at each point in time. This CPU time is tracked independently for CPs and zIIPs.

- The size of a Job may change at different points in time, due to changes in its CPU consumption (e.g. a Job may change its Workload Component attribute from 'Small' to 'Tiny', because it used less CPU).
- CICS values may differ from Supervisor Scope values for a Workload Component, due to only a subset of Jobs applying to this Scope.
- If the CICS Cpu Time accounts for a high percentage of the Supervisor CPU Time for a Workload Component, then Supervisor Activities may be applied to CICS Activities (e.g. Dispatches per CICS Transaction).

**Cpu Time Per Transaction** (Group: Detail) - This compound Activity tracks the average CPU time consumed per CICS task, at each point in time. This CPU Time is normalized and includes CP plus zIIP time.

- See Supervisor Scope 'CPU Time per Dispatch' for possible causes for (transient) Activity value changes.
- For transactions that are dependent upon DB2, increased buffer pool concurrency may contribute to Activity value increases.
- Changes in worst offending jobname or transaction name over time (as seen on graph hover text), may also account for 'Cpu Time Per Transaction' value changes.

**Jobs Contributing** (Group: Detail) - This simple Activity tracks the number of Jobs that contributed to this CICS Workload Component, at each point in time.

- CICS values may differ from Supervisor Scope values for a Workload Component, due to only a subset of Jobs applying to this Scope.
- This Activity is included in graph hover text for many Activities.
- Changes in the number of Jobs Contributing to a Workload Component, may impact the current Activity's graphed values, at the point of change.

**Response Time Per Transaction** (Group: Detail) - This compound Activity tracks the average Response Time per CICS task, at each point in time.

- Many factors may contribute to the elongation of Response Times. These include delays incurred by services the CICS task is dependent upon, and / or the amount of contention for resources it consumes.
- High competition for access to CPU may negatively impact Response Times. Consider consolidating a large number of CICS 'Tiny' jobs performing the same work into fewer CICS 'Small' jobs, where applicable.
- Changes in worst offending jobname / transaction name over time (as seen on graph hover text), may also account for changes in the 'Response Time Per Transaction'.

**Transactions** (Group: Detail) - This simple Activity tracks the number completed CICS tasks across both CPs and zIIPs, at each point in time.

# Need Help?

Clients are encouraged to analyze Correlator data through the Navigator to identify problem areas, determine a corrective action, and take that action - iterate. The Navigator provides a direct feedback loop, that enables clients to determine the progress toward meeting their availability or performance goals.

If a client feels they are ineffective in determining the corrective action to be taken, contact IBM Lab Services for education or analysis assistance to use the client's Interacting Trends data to provide client specific recommendations - iterate.