

ABA Configuration Tuning

Michael Sydor

SERVICE ASSURANCE - APM

PRE-WORK	4
Your Preparation	4
Setting Expectations and Scope	4
Day	4
Week	4
Understanding the ABA Lifecycle	4
EXECUTION	5
Access	5
Activities	5
ABA Tuning	5
Why does the default ABA configuration FAIL?	5
KPI Analysis And Collection	7
Generating the ABA Configuration	9
Default Configuration and validation	11
REPORTING	15
Screenshots	15
Wrap-up Presentation	15
REFERENCES	16
PDFs	16
PDFs	16
Books	16
ABOUT THE AUTHOR	16

Copyright ©2014 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. This document is for your informational purposes only. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "as is" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages.

Scope

A *Configuration Tuning* is a structured review of the settings and performance, with the goal of optimizing the capability of various components of your APM solution. This Configuration Tuning is focused on the ABA component of the APM solution.

The processes discussed are appropriate for client practitioners, pre-sales and services consultants. You will want to have a good understanding of using the APM Workstation: navigation, searches and management module editing. You should also have good skills in keeping track of your work with NotePad and SpreadSheet.

Duration

A Configuration Tuning may be completed in a few hours or may take a week or more, depending on the frequency that you can introduce configuration changes to the production environment, after some allowance for the metric stream to be re-established. As the ABA component is a subscriber to the metrics stream, production changes should be straightforward as no application or agent configuration is being impacted. Earlier versions may require a restart of the CA APM MOM but this is no longer necessary from APM 9.6 as the `Analytics.properties` files, which contains the ABA configuration, is now a 'hot' property.

Benefits

Completing a Configuration Tuning is important to the long-term success of an APM initiative and to ensure the intended operation of the ABA component. Tuning of the APM configuration is often a significant gap which you can help the customer overcome.

- Confirm correct operation and potential solution capacity
- Uncover significant KPIs for the client application environment
- Tailor the ABA Configuration specific to the client application environment
- Understand how to reliably employ the new visibility in diagnosing application performance problems

PRE-WORK

Your Preparation

Currently, the primary reference for tuning the ABA product is this document. This should change in the future as the 'CookBook' is integrated within the product documentation. You will also need to operate the CA APM Workstation Investigator to assess the APM and ABA operational health, identify KPIs that will be added to the ABA Configuration, and operate the CA APM WebView Console to interact with the information provided by ABA. This too will also change as the Workstation/WebView change in functionality, over future releases.

Interpreting performance data takes practice. It is actually a form of triage which is discussed in Chapter 14 – Triage with Single Metrics in **APM Best Practices – Realizing Application Performance Management**. You will find specific details of ABA performance situations in this document, as they become available.

The identification of KPIs is a manual step and may already be underway. You will need those KPIs in a simple text file which will then be processed by a Python script to generate the new ABA configuration. This result is then manually pasted into the `Analytics.properties` files.

If a KPI practice is not yet established, you will need to allow 1-2 hours to collect KPIs from a correctly operating environment.

If the APM environment is not operating correctly, then completing the KPI analysis will be a very painful process. This is because the collection of KPIs requires broad queries against the stored metrics. If the APM is performance challenged, completing these queries will take a long time and can also cause the APM Workstation to become unstable, requiring reboot of the workstation. When things are difficult, your KPI analysis will also suffer because you will start to take shortcuts to mitigate the pain.

Please consider an *APM HealthCheck* (see Reference Materials) prior to undertaking an ABA Configuration Tuning if there are any concerns for the environment.

Setting Expectations and Scope

Day

The ABA Configuration Tuning is generally completed in about 2-8 hours. The client will have been operating an APM solution for at least 6 months, prior to the introduction of CA APM ABA. The APM environment should be stable and operating within its primary KPIs (Duration of Harvest, SmartStor and GC are each <1500 msec – without exception!) Generating a new ABA configuration involves some manual steps and external tools but will progress very quickly provided that the pre-work is completed.

Week

If you are tasked with doing a Configuration Tuning over five days, it will usually be because the environment is medium to large scale. A couple of APM clusters and you will have a lot of data to review and perhaps a lot of adjustments to schedule in both production and pre-production environments. Production changes are usually done after normal business hours. This is not necessary for the APM environment but sometimes agent configurations will need adjustment and this is what takes time. Specifically, time for testing of the proposed changes, deployment and validation of the new agent configurations, will be the time-consuming steps.

Understanding the ABA Lifecycle

ABA needs about two weeks to establish patterns with the applications that are being monitored. Candidate metrics that are generated infrequently, especially batch operations and almost everything that happens in QA testing – these will make very poor candidates and result in excessive anomaly detection. Any changes in the ABA configuration first undergo syntax checking, and then the number of metrics subscribed will begin to change. Anomalies may start appearing but these should be ignored until the application patterns are established.

If your client wants to evaluate ABA in a pre-production environment then you will need to set expectations about the remote chance of seeing anomalies of any significance. You will see a lot of anomalies but they will all be due to the artifact introduced by having applications in limited testing – and then quiet or disabled. It's just not going to be effective. The concern will be for placing an untested component into Production. So any pilot pre-production will be limited to testing the setup of the ABA server and the delivery of metrics from the MOM to the ABA component. You are not going to see any significant anomaly detection until ABA is seeing a production load AND having an appropriate configuration – which is what this document is setting the stage for.

The end result is that changing the ABA configuration is not something to undertake casually. You really want to understand why your current configuration is not working and what your applications really generate (in terms of KPIs), in order to arrive at a useful change. This is the path we took in order to optimize the ABA configuration, for the various environments we tried during our evaluation period. And you will find some of those details in this document – so you don't have to re-invent the wheel!

For more details of how ABA works and is installed and configured, please see page 10 of the *ABA install and Configuration* guide.

EXECUTION

Access

Before going on-site you need to confirm that you will have someone to access the APM Workstation. It is very rare that you will be able to login to a production system, let alone hook your laptop into the operational network. What you may have had latitude to do during a pilot is simply not allowed with production.

In addition to APM Workstation access, you will also want to confirm that you will have PowerPoint and a browser with connectivity to the outside. You will need to screenshot and annotate – PowerPoint is best for this. You can also use Word or something equivalent. Screenshot everything! This is all you will have on which to base your analysis and recommendations. You will also be transferring files (ppt and pdfs, typically) which you can do via a browser-based (webmail.ca.com) email session. We find that it is very rare that even a USB port is open on any machine that touches the production network. Be prepared!

Activities

In this section we will discuss how to plan your time, either remote or on-site.

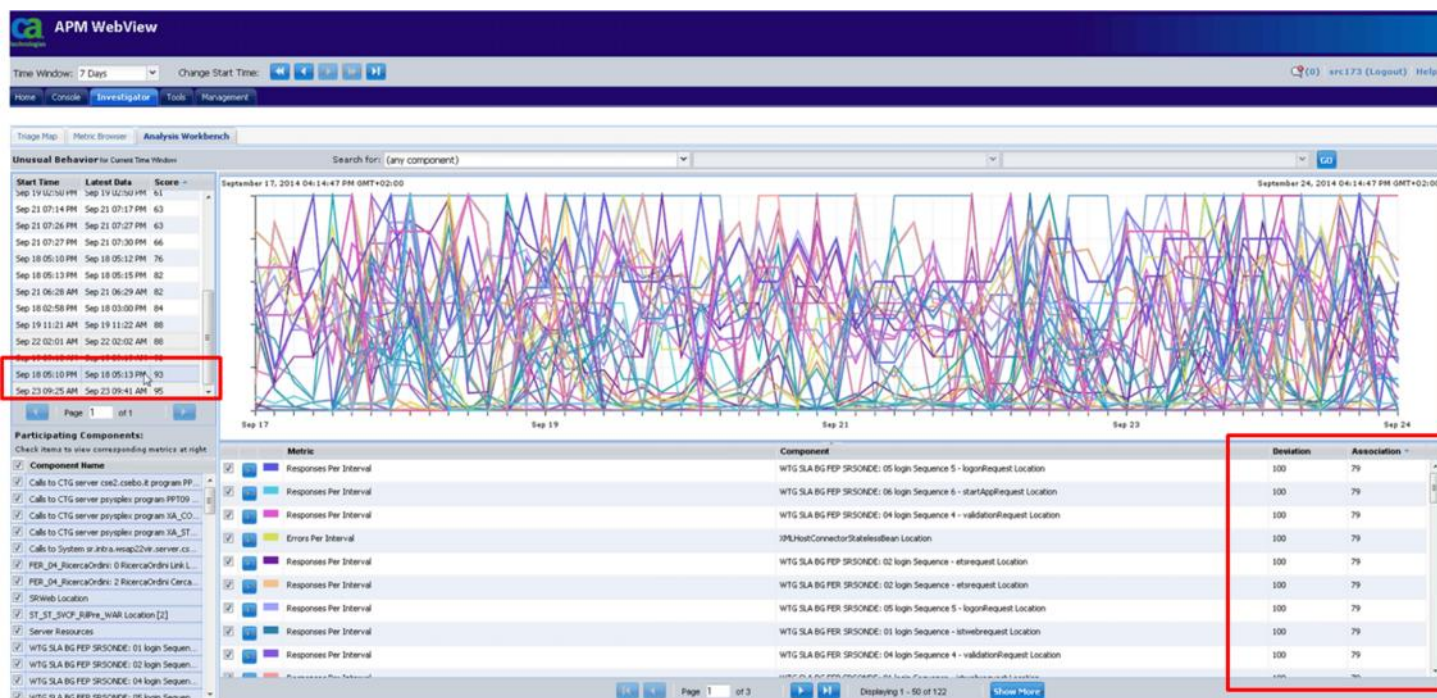
ABA Tuning

Successful ABA tuning depends on a good familiarity with the CA APM Workstation and results are currently best achieved via the thick client. Please look to the Workstation Guide for guidance on using the APM Workstation.

The topic of KPIs (Key Performance Indicators) is fundamental to a successful APM practice. Please look to the “CookBook – Application Audit” as well as Chapter 12 of “APM best practices” for detailed discussion and examples. We will be using only the most basic elements of this essential APM process for our ABA tuning but you will find a much richer use of these techniques if you continue your study.

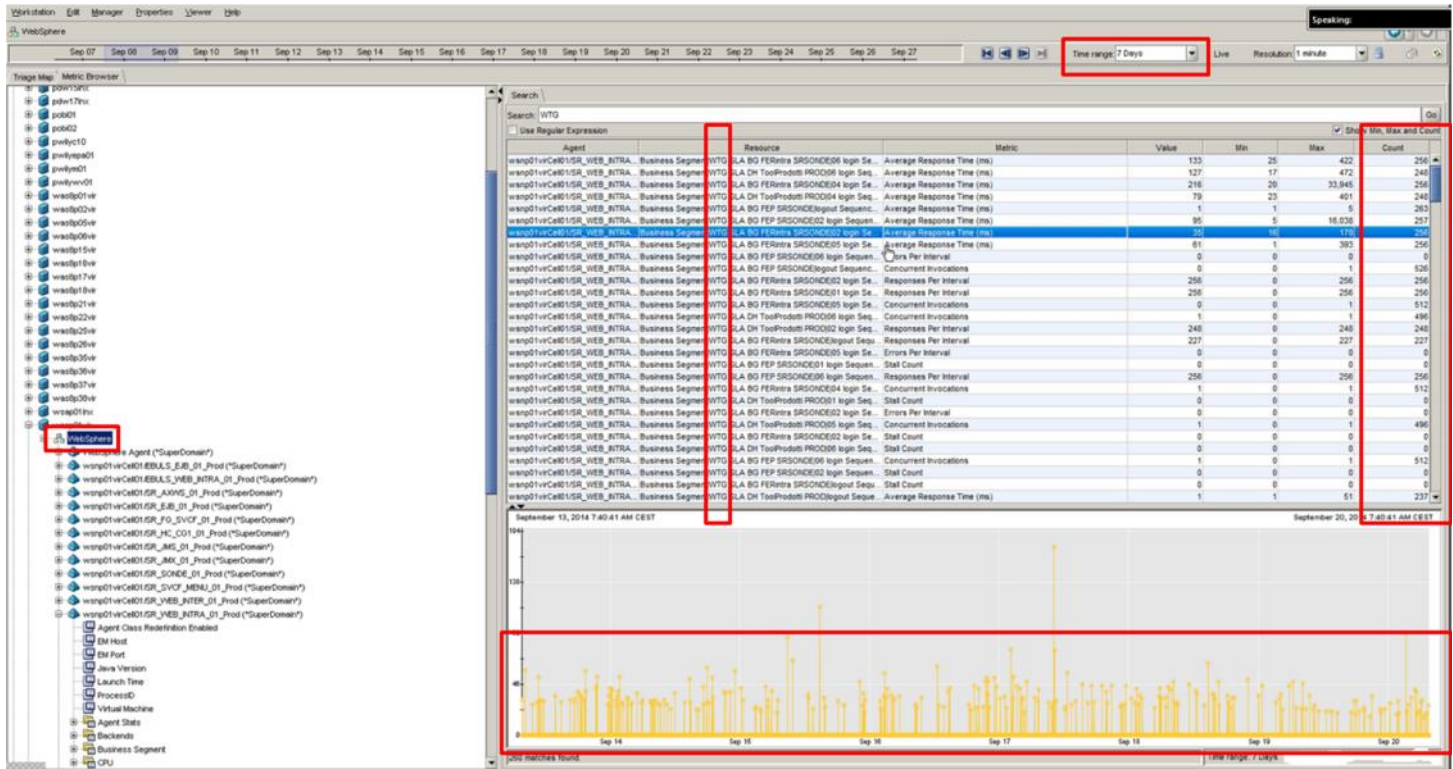
WHY DOES THE DEFAULT ABA CONFIGURATION FAIL?

The default configuration fails because it generates excessive and inaccurate anomalies and this is due to both the wrong types of metrics being sent to ABA, as well as the volume of some of those metrics. Here is an example of an anomaly that is otherwise uncorrelated with an actual production performance problem:



You can see that we have a high score (93), a component count of (122) beginning with “WTG” and deviation of (100) units. Looks like trouble!

If we go to the APM Workstation Investigator, which is necessary because it can show the “Min, Max and Count”, we can easily explain why this is happening.



Our scope, on the left, is the entire application cluster, so we catch all of the “WTG” metrics. We are thus searching on “WTG” (see Search field) and find, to no surprise - all of the “WTG” metrics. These are the Front-End components that correspond to the CEM transactions that the default ABA Configuration is looking for. We are looking at a (7) day historical range, including the date of the anomaly and basically, there is nothing going on here!

Really! - nothing going on. These “WTG” transactions have a weekly frequency or invocation rate of 250-500 per week. Looking at the graph, we see that the transaction is in fact running very sporadically. What these means for ABA is that it tends to score these frequent appearances and disappearances as “anomalies”.

The threshold for a meaningful KPI, based on experience to date (DEC 2014), is 10,000 invocations per week. Metrics that are less than 10,000 invocations (counts) are not consistent enough to anchor an operational threshold. There are simply too many gaps in the data stream and this leads ABA to consider the gaps as an anomaly. Metrics that are greater than 10,000 invocations are good candidate KPIs in that they should typically not have any gaps, at that volume.

If we apply the KPI technique, and focus on “Average”, for that same search scope, we get a very different insight into what is *really* significant for the application.

Now it is entirely possible for a metric to have a Count of >10,000 and still have too many gaps to be useful for ABA. Maybe all of the invocations happen on a Monday, with nothing else for the week? Maybe it is really active for 1-2 hours every day and at all other times inactive? There will be such odd cases and if you look at the candidate metrics, with 1 minute resolution, the graph will reveal that they are consistent – or not. We suggest you avoid sending those metrics to ABA if you encounter some. And certainly, if you have a component generating a large amount of anomalies, you can quickly check if it is related to this volume situation, or not.

Process Overview

If you know which applications are significant, you can limit the scope to those applications. If you don't have any preference, doing the search at the SuperDomain will also be effective.

Note that a SuperDomain search will take a significant time. If your cluster (or Collector) is already having performance problems, the queries will take 10-15 minutes to complete. Any search result will be limited to 500 metrics returned to the workstation, which for the purposes of KPI identification, completely suits our needs. This is where the “Count” comes in – we simply sort on the ‘Count’ to ensure that we are looking at the most significant metrics, in terms of invocation rate for those components. This sorting is carried out on the APM MOM, prior to the set of 500 metrics being delivered to the APM Workstation. The ‘count’ is evaluated over a (7) day historical view. A reliable ‘rule-of-thumb’ is that a KPI is significant if it has at least a count of 10,000 over a 7 day period.

We are today collecting (5) types of metrics, using the following target keywords:

“Average”
“Responses”
“Errors Per Interval”
“Stall Count”
“Stalled”

You can of course search on any type of metric you find is significant. Typically, suitable KPIs will not exceed more than 1% of your total metrics. So if you follow our guidelines, you will almost certainly get the bulk of your KPIs, before you even consider other potential sources.

Some of the alternate sources are SysView metrics (from Cross-enterprise Transaction Correlation) and SAP. Any Custom metrics that follow the typical metric units will already be evaluated with the target keywords above.

When you have searched on a keyword, you will then select all of the results and then copy, then paste into a Notepad window.

Notepad is the preferred tool to capture the search result as it will automatically preserve the <TAB> delimiters – and this will improve the parsing accuracy when the ABA configuration is created. This is known today NOT to be effective on Apple Macintosh products. If the notepad functionality is not cooperating, please consider using a spreadsheet to receive the copy/paste. Be sure to save the spreadsheet as a test file, for subsequent processing by the python utility.

Detailed Procedure

1. Open the APM Workstation, PowerPoint and NotePad. Use a spreadsheet if your NotePad/textEditor is dropping the <TAB>.
 - a. PowerPoint will be used to capture screenshots of your metrics capture, documenting the number of metrics matched and the number of metrics viewed. The viewed metrics will be limited – this is not a problem – but it is interesting to discuss how much the search considers and what is appropriate for the KPI candidates.
2. Open an Investigator session in the APM Workstation and select the Metric Browser tab. Navigate to the SuperDomain and select that level of the metric hierarchy.
3. Click the checkbox for “show Min, Max and Count”.
4. Toggle the <Live> button so that you can select a historical range of 7 days.
5. Using the <Search Tab>, enter one of the keywords [Average, Responses, Errors Per Interval, Stall Count, Stalled]
6. When control of the workstation returns, the bottom of the workstation window will change from “searching” to a report of the number of metrics matched and the number viewed.
7. Select the top row, then <CNTRL><A> to select all of the metrics. The graph will populate and when this is finished, <CNTRL><C> to copy the selection.
8. Paste the selection into the NotePad window and immediately save the file as “search_average.txt”, changing the name of the file to match the search term. The naming is not critical but each file must be distinct.
9. Now take a screenshot of the Investigator window and paste into PowerPoint. Save this file with your company name and which environment was being analyzed, such as “CA-Inc_production_KPIs.ppt”
10. Repeat steps 5-9 until you have collected all of the candidate KPIs into their respective *.txt files, and have all of the screenshots in your “company_environment_KPIs.ppt”
11. If you are participating in the beta program, collected together all of the *.txt, *.ppt and your current Analytics.properties file, and return to CA for processing. You will receive a new Analytics_broad.properties and Analytics_narrow.properties, which will replace your current ABA configuration. You will receive instructions on managing the various Analytics.properties files with the new configurations.

The following screenshots summarize the workstation selections:

Investigator - CA Inroscope Workstation

Workstation Edit Manager Properties Viewer Help

SuperDomain

Time range: 7 Days Live Resolution: 168 minutes

Triage Map Metric Browser

Search: Average Use Regular Expression

✓ Show Min, Max and Count

SORT on this column

370959 matches found. Limiting viewable data to 500 matches.

Host	Process	Agent	Resource	Metric	Value	Min	Max	Count
Custom Metric Host (Virtual)	Custom Metric Process (Virtual)	Custom Business Application Agent (Virtual)	By Fronte...	Average	1	0	5,883	291,245
uxpappac23	awdntmprod	awdntmprod	Backendst...	Average	1	0	292	1050288
uxpappac25	awdfsp1prod	awdfsp1prod	Backendst...	Average	0	0	272	284438
uxpappac20	awdfscnprod	awdfscnprod	Backendst...	Average	0	0	37	231844
uxpappac27	awdfcnprod	awdfcnprod	Frontendst...	Average	1	0	240	194386
uxpappac22	awdfatgprod	awdfatgprod	Frontendst...	Average	0	0	1	143069
uxpappac20	awdfhannprod	awdfhannprod	Backendst...	Average	1	0	277	121960
uxpappac20	awdfnvcprod	awdfnvcprod	Backendst...	Average	7	1	1,356	94246
uxpappac24	awdfhivprod	awdfhivprod	Backendst...	Average	3	0	5,644	91868
uxpappac20	awdbukcprod	awdbukcprod	ServletFi...	Average	2	0	5,247	91632
uxpappac23	awdfscnprod	awdfscnprod	JMXJboss...	Average	0	0	0	80578
uxpappac24	awdfsp2prod	awdfsp2prod	JMXJboss...	Average	0	0	1,139	80545
uxpappac27	awdfntmprod	awdfntmprod	JMXJboss...	Average	0	0	0	80543
uxpappac27	awdfsp1prod	awdfsp1prod	AWC(JAFT...	Average	0	0	0	75945
uxpappac20	awdfntmprod	awdfntmprod	Backendst...	Average	0	0	280	64158
uxpappac20	awdfhlmprod	awdfhlmprod	Frontendst...	Average	0	0	5,469	63668
uxpappac25	awdfsp2prod	awdfsp2prod	Frontendst...	Average	1	0	451	63667
uxpappac20	awdfntmprod	awdfntmprod	EJB(Sessi...	Average	147	0	7,006	41743
uxpappac23	awdfhannprod	awdfhannprod	Backendst...	Average	2	0	254	39604
wdc-ema02	bmcaacdp	bmcaacdp	JMXJavaL...	SystemLo...	3	0	3	38936
uxpappac26	awdfhivprod	awdfhivprod	Heuristics...	average r...	2	1	2	38132
uxpappac23	awdbadonprod	awdbadonprod	Heuristics...	average r...	2	1	2	35088
uxpappac21	awdfuxprod	awdfuxprod	EJB(Sessi...	Average	133	50	79,185	34745
uxpappac23	awdfhagprod	awdfhagprod	Backendst...	Average	3	2	252	32298
uxpappac23	awdfhannprod	awdfhannprod	Heuristics...	average r...	2	1	2	28744
uxpappac20	awdfscnprod	awdfscnprod	Heuristics...	average r...	2	1	2	28424
uxpappac23	awdfhannprod	awdfhannprod	Heuristics...	average r...	2	1	2	28404
uxpappac23	awdfntmprod	awdfntmprod	Backendst...	Average	0	0	2	26234
uxpappac21	awdfntmprod	awdfntmprod	EJB(Sessi...	Average	2	1	296	25628
uxpappac23	awdbukcprod	awdbukcprod	Backendst...	Average	1	0	107	24912
uxpappac20	awdfhannprod	awdfhannprod	Backendst...	Average	0	0	1	24261
Custom Metric Host (Virtual)	Custom Metric Process (Virtual)	Custom Business Application Agent (Virtual)	By Fronte...	Average	130	4	8,732	23449
uxpappac26	awdfndcprod	awdfndcprod	EJB(Mess...	Average	281	25	13,591	23099
Custom Metric Host (Virtual)	Custom Metric Process (Virtual)	Custom Business Application Agent (Virtual)	By Fronte...	Average	4	2	210	22052
uxpappac27	awdfsp1prod	awdfsp1prod	Frontendst...	Average	0	0	452	21903
uxpappac23	awdfhannprod	awdfhannprod	Backendst...	Average	18	2	3,824	21738
uxpappac23	awdfatgprod	awdfatgprod	Frontendst...	Average	16	1	1,111	21627
uxpappac22	awdfnvcprod	awdfnvcprod	Frontendst...	Average	0	0	28	20943
uxpappac25	awdfsp2prod	awdfsp2prod	Heuristics...	average r...	2	1	2	20937
Custom Metric Host (Virtual)	Custom Metric Process (Virtual)	Custom Metric Agent (Virtual)	AlertsHar...	FrontEnd2...	1	0	1	20937
uxpappac21	awdfnvcprod	awdfnvcprod	Heuristics...	average r...	1	1	1	20937

Refinement

While the technique described will result in a much lower number of metrics, while still focused on the more significant metrics, there are still additional refinements to consider.

For the Response Time KPIs (Average) you should also make an effort to avoid any candidates that have less than 10,000 invocations per week (7 days). Metric frequency in the range of 10-10,000 invocations is already known to generate a large number of uncorrelated anomalies – effectively crying wolf, when no wolf exists! This is really easy to do: simply don't copy any metric that has less than 10,000 invocations (the Count column).

This will also be the case with the Capacity KPI (Responses Per Interval) – simply don't copy any metric that has less than 10,000 invocations (the Count column).

For the stalls and errors, this guideline would not apply as these values will be zero, unless there is a problem. The invocation rate should not matter.

For other metrics that you might consider, you'll have to make your own judgment call. But if you find a configuration that gives strong correlation between anomalies and production incidents – please don't keep it a secret! Post your findings on the Community site and everyone can benefit.

GENERATING THE ABA CONFIGURATION

There is a useful background discussion on customizing the ABA Configuration, which may be found on page 29 of the *ABA install and Configuration* guide. This process introduced in this cookbook is somewhat different. You may want to consider both approaches at some time, for different situations.

Using the process for identifying KPIs, discussed in the prior section, you will have one or more test files of *candidate KPIs*. You can generate either a "broad" or "narrow" configuration. It's really quick, so generate both of them and try the *broad* one first. I think that the *narrow* will be the best but I really don't have any evidence (yet) either way.

The *broad* configuration uses the component name and a wildcard for the metric name/type, such as:

```
Component_name: (.* )
```

This will match up to (5) metric types for a typical component. While this may be overly broad we were of the mind to stay close to the original design goal of having fewer metric expressions to gather a fair amount of metrics.

The *narrow* configuration will only generate a regex for the specific metric. So if you were going after response time metrics, then the following would be generated:

Component_name:Average Response Time

This represents the most efficient selection and precisely aligns with the intent of the KPI technique, and is expected to offer the best ABA operation – but this is currently conjecture – we haven’t been able to gather enough results from our beta sites to say for certain.

So here is the command line:

```
$ ./build_regex.py -list sap_list -output sap_broad -exceptions sap_broad_exceptions
...writing to sap_broad
...writing exceptions to sap_broad_exceptions
...processing ['./CA-GIS_beta/SAP_average.txt', ''] Index:: 0 Lines Written:: 0
...processing ['./CA-GIS_beta/SAP_errors.txt', ''] Index:: 436 Lines Written:: 2616
...processing ['./CA-GIS_beta/SAP_responses.txt', ''] Index:: 907 Lines Written:: 5442
...processing ['./CA-GIS_beta/SAP_specific.txt', ''] Index:: 1319 Lines Written:: 7914
...processing ['./CA-GIS_beta/SAP_stall_count.txt', ''] Index:: 1323 Lines Written:: 7938
...processing ['./CA-GIS_beta/SAP_stalled.txt', ''] Index:: 1816 Lines Written:: 10896
>>> Evaluated 2505 expressions and got 2316 valid with 189 invalid.
>>> Next index is 2316 and 13896 total lines were written to output file.
```

With the new configs, in this case sap_broad, copy paste the goodies into your Analytics.properties file and save it with a unique name. This will allow you to move quickly between original, broad and narrow configurations.

For the invalid configurations, you can try and resolve these manually, using the Search tab and checking the "use regex" box. I just ran out of time to figure out why the otherwise valid regex is not palatable to the MOM and ABA.

You might find that you already have a pretty good ABA configuration and want to preserve that. So find out the last index that you used, add one and then you can use that to set the starting point for all the ABA config that you are about to generate. Here is that command line variant, this time working with a single file:

```
$ ./build_regex.py -file SAP_average.txt -index 23 -output sap_broad -exceptions sap_broad_exceptions
...writing to sap_broad
...writing exceptions to sap_broad_exceptions
...processing ['SAP_average.txt'] Index:: 23 Lines Written:: 0
>>> Evaluated 501 expressions and got 436 valid with 65 invalid.
>>> Next index is 459 and 2616 total lines were written to output file.
```

You can also find all of the current capabilities by using the -help or -h parameter:

```
$ ./build_regex.py -h
usage: build_regex.py [-h] [-list LIST] [-output OUTPUT]
[-exceptions EXCEPTIONS] [-depth {broad,narrow}]
[-debug {none,low,high,all}] [-file FILE] [-index INDEX]
optional arguments:
  -h, --help show this help message and exit
  -list LIST supplying a list of filenames of candidate metrics to
        be processed into regex suitable for ABA
  -output OUTPUT file to capture regex suitable for ABA
  -exceptions EXCEPTIONS
        file to capture any unexpected metric types
  -depth {broad,narrow}
        controls quantity of regex (more/less metrics)
  -debug {none,low,high,all}
        controls debug info level into output file
  -file FILE supplying a single file of candidate metrics to
        process
  -index INDEX set the starting index for the regular expressions
```

DEFAULT CONFIGURATION AND VALIDATION

Every application environment is different. In this section we will review exactly how you can evaluate the default ABA configuration to understand why it might work and why it might not be effective. You should also consider these techniques if you are using regular expressions as part of your customization strategy. As you will find in the previous section, our technique for generating a new ABA configuration generates a large number of expressions. If your applications are better understood, you may likely find it possible to devise a smaller number of expressions. You can use the techniques here to validate exactly what those regex will produce and be able to confirm that they meet the guidelines (>10,000 invocation per week) we have established.

The default ABA configuration contains (4) regular expressions or *regex* that establish which metrics from the APM environment will be sent to the ABA component. These are contained in the Analytics.properties file. Here is one of the regex:

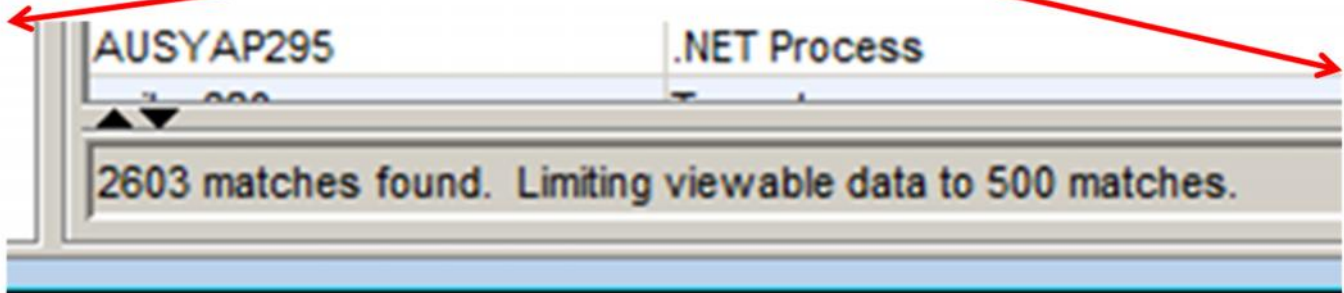
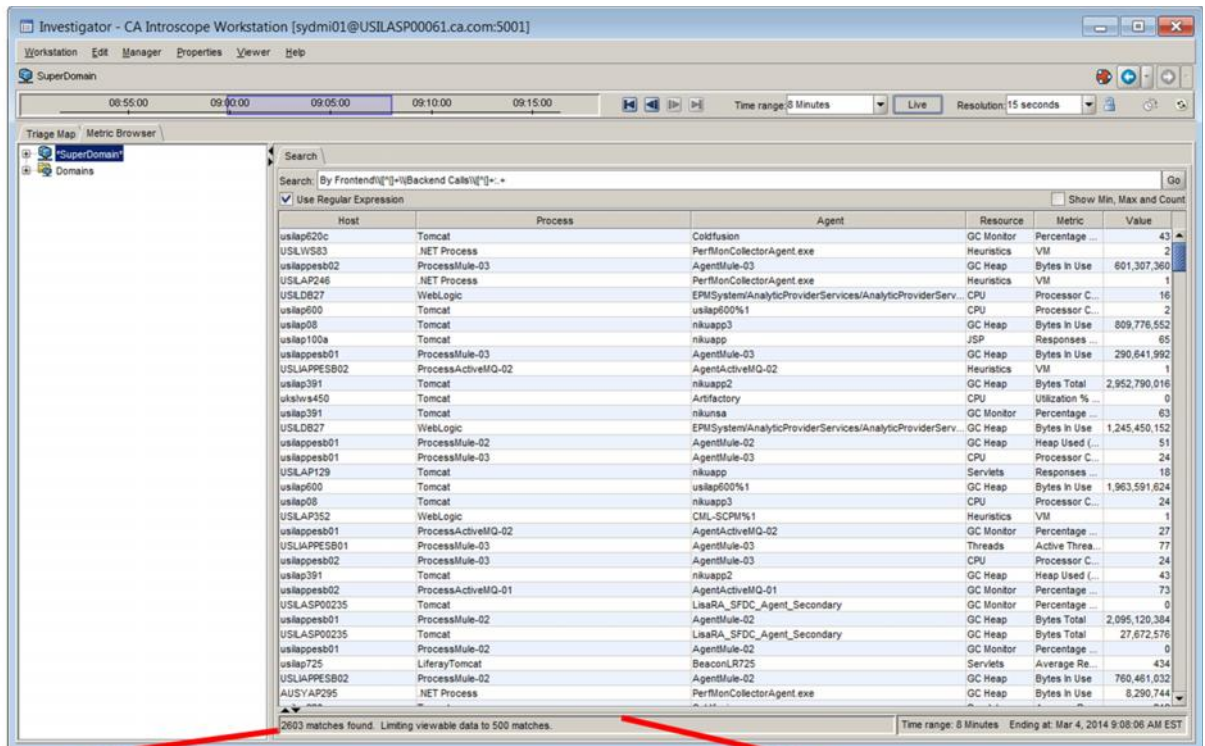
```
analytics.metricfeed.process.1 = Custom Metric Host (Virtual) \\Custom Metric Process
(Virtual)\\Custom Business Application Agent (Virtual)
analytics.metricfeed.metric.1 = By Frontend\\[^[^]]+\\Backend Calls\\[^[^]]+\\.+.+
```

To evaluate exactly what this will match, in your environment, open the APM Workstation Investigator, select the Super Domain [1] and then select the Search tab, and copy/paste the regex [2] (highlighted in **green** above) as follows:



Also note the checkbox [3] where you enable the “Use Regular Expression”.

When this regex is evaluated, you will then get a count of all the metrics that match, and a display of the top 500. You can then sort the various fields to focus on metric types of interest.



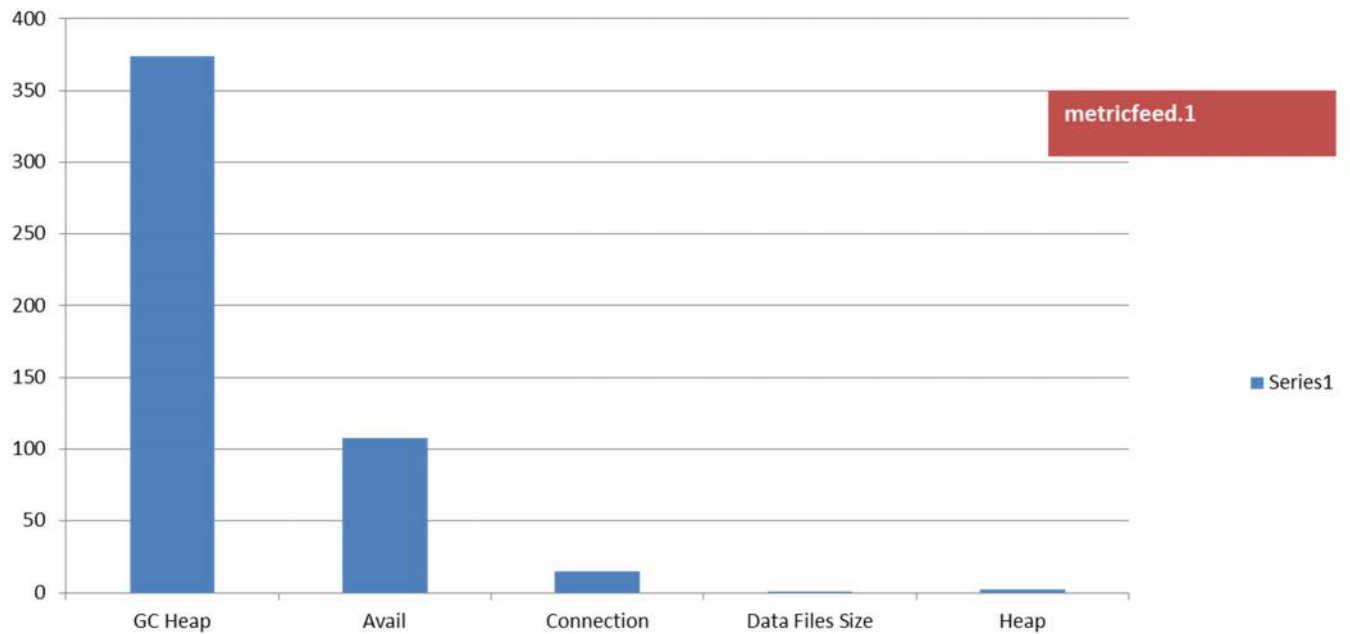
So we checked each of the default ABA regex in this fashion, and gathered statistics on each of the metric types encountered. This is a little tedious to complete but feel free to reproduce the analysis as you like. Here is what we found:

Study 1	What is the overlap between ABA metric RegEx and baseline process						
APM Environment		Prod - non-SAP		usilasp00061 / 5001			
		Total Metrics		262082			
ABA				Baseline			
Regex	# matching	#displayed	% of available	Query	# matching	#displayed	% of available
0	0			Average	43625	500	16.65%
1	2547	500	0.97%				
2	2605	500	0.99%				
3	2580	500	0.98%				

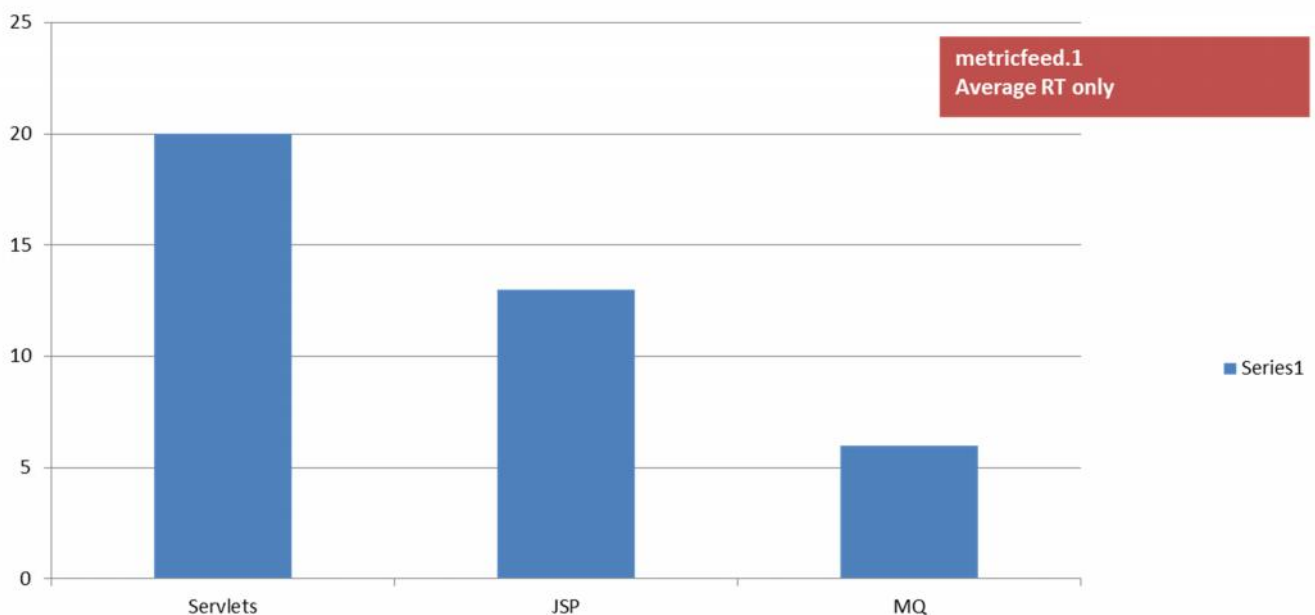
We found that the default ABA configuration was matching about 1% of the available metrics. As a control we used the baseline technique, or "KPI Analysis", that is part of the APM Best Practices, and found that it considered almost 17% of the available metrics, including 100% of the response times metrics that are often the best candidate for a performance KPI. This is not really an "apples to apples" comparison but it reveals a specific advantage of the baseline technique in establishing a pool of metrics to look at. In either case, we would limit our analysis to the most significant candidates of the search pool; the top 500.

Next, we would tabulate the details of every metric selected and then apply additional criteria to ensure that the candidate metrics were of good predictive quality.

Here are the results for metricfeed.1:

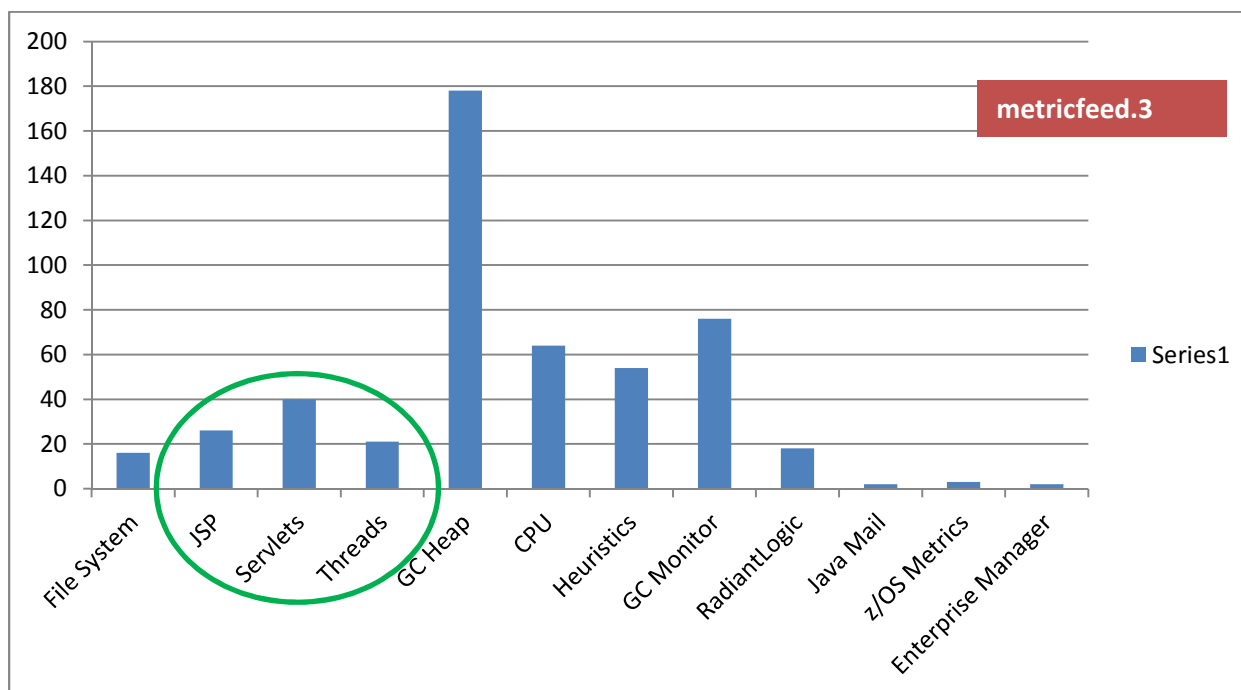


Here we see that the bulk of the metrics matching are not especially useful for triage. GC Heap is generally unreliable, except when it is in the extreme. Availability is not predictive of performance – you are either up or you are down – and there are simple ways to establish that situation without resorting to ABA. If we instead focus on the response time metrics, we can get the following categories:



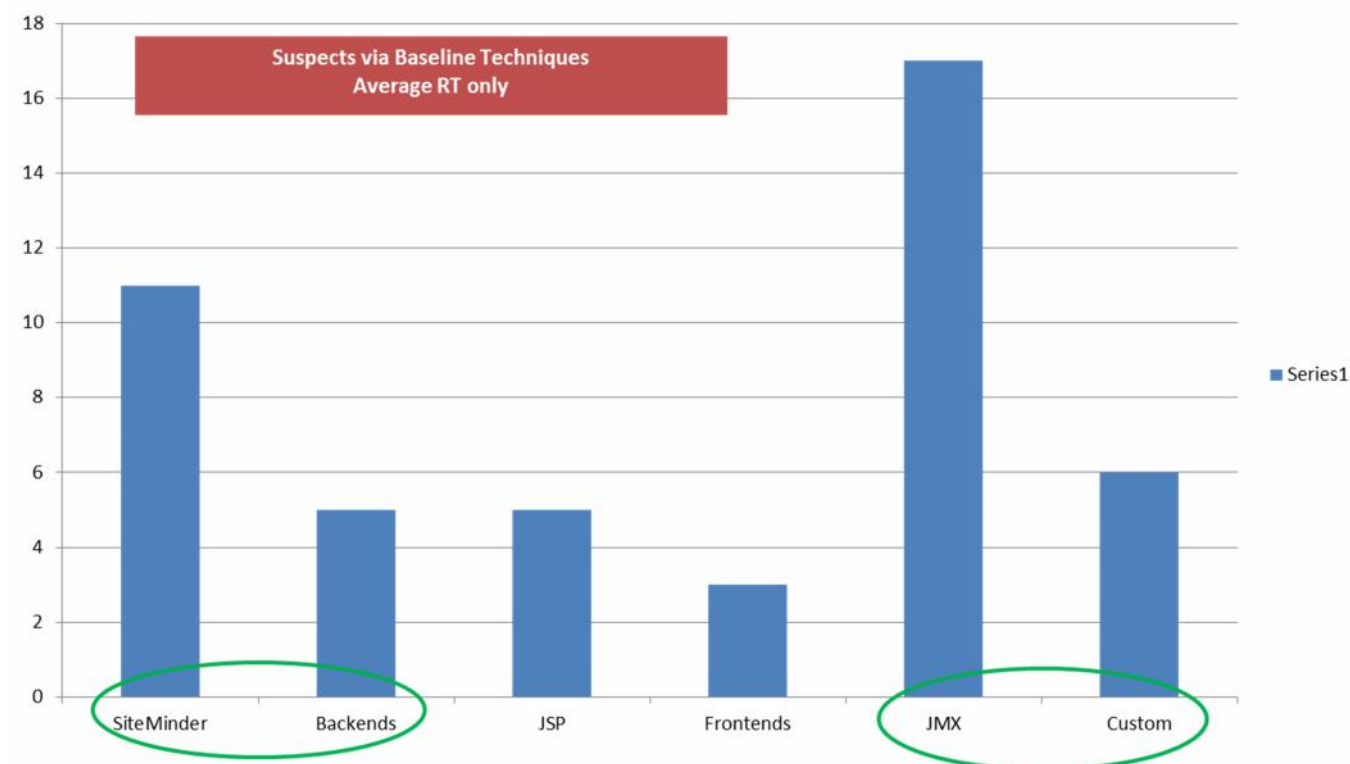
Here we have solid performance predictors: Servlets, JSPs and some MQ. Not very many – only 37/2600 == 1.4% - but these are generally accepted as predictive of performance, once appropriate thresholds are established, which is the very job ABA needs to perform. The problem for ABA is the other 98% of the metrics stream which potentially introduce all kinds of potential anomaly alerts.

With metricfeed.3 the situation improves:



Here we have a better variety of metric types, including the 'usual suspect': JSP, Servlets and Threads. But also some other components like Radiant Logic and z/OS. We also have a bit of overlap (GC heap) as well as some inappropriate (Heuristics, Enterprise Manager) which are not part of the application landscape. This contributes $87/500 == 17.4\%$ generally known as useful for performance prediction and triage.

Next we will consider the baseline technique:



Here we find four interesting additions which the default ABA regex have missed, in addition to the two we would expect (JSP, Servlets). And they are *killer* candidate KPIs. SiteMinder, which is the authentication system, Backend metrics, of which none were identified for ABA. JMX metrics, which normally we considered second class because they are often static configuration values. Here we are focused on response time values so all of these JMX are solid contributors. And then the Custom metrics, which are extension pbds (probe builder

directives) and totally unique to this operating environment – but metrics that we establish very much on purpose – so they should always have good value.

Now this also represents about 0.1% of the total metrics considered (47/43625) but every one is a solid candidate for predicting performance. And we have a much broader scope for our search.

So now that you know about the other sources of valuable metrics, you could easily create a regex to catch them and add them to the ABA configuration. And that is very much our plan. But there is one more important element to the KPI Analysis – it specifically focuses on metrics that have a significant volume, excluding those that have a lesser volume. The magic number is 10,000 invocations per week but what it insures is that the data stream is consistent throughout the week. Lesser volumes tend to produce gaps in the data, as if the application went away (became unavailable), and this means more spurious anomalies. The default ABA regex do not make any distinction by volume about the candidates that they match.

So this becomes the strategy for a more effective ABA configuration. Focus on significant (known to be useful for triage) KPIs. Narrow those candidates to those with sufficient volume (>10,000 invocations per week) and only use a regex that will capture those specific candidate KPIs.

REPORTING

Screenshots

Screenshots are always more efficient then writing down enough details to actually reproduce the view you are currently interested in. It also makes it completely simple to annotate and later communicate and mentor the APM team. It also makes it easy to get help, via the WIU, via email. Everybody loves a puzzle and a picture says a lot – compared to your under-pressured-written-description of the problem. The APM workstation today also presents a couple of features to copy just the graphs and this will make for a neater presentation or report, as desired.

Wrap-up Presentation

You will want to cover (at minimum) the following topics during your wrap-up:

Slide 1 – What was done

- Configuration Tuning of ABA
- Which environments (for which ABA is deployed)
- Which EM servers participating
- Any impediments
- Any EM or ABA health issues

Slide 2 – Types of KPIs Considered

- Metric Types
- Number of regex resulting
- Change in #metrics from EM to ABA
- # KPIs selected in each category vs. number of metrics matching search criteria

Additional Slides – showing details of specific finding or anomalies that correlate with production incidents

REFERENCES

PDFs

APM Application Behavior Analytics Installation and Configuration Guide 9.6.pdf – Available of the CA Support Site, Application Performance Management Bookshelf

CookBook – APM HealthCheck.pdf – Available on the CA Community Site

CookBook – Application Audit.pdf – Available on the CA Community Site

PDFs

Build_regex.py – Available on the CA Community Site

Books

APM Best Practices – Realizing Application Performance Management ISBN 978-1-4302-3141-7 – Available on Amazon.com or APress.com in print or electronic formats. Also available on Safari.com (subscription) in electronic format.

Electronic formats are full color!

ABOUT THE AUTHOR

Michael Sydor is an Engineering Services Architect specializing in Best Practices for APM. He advises and leads client teams to establish their own APM disciplines to deliver effective triage and manage performance across the application lifecycle. Michael is also the author of “Application Performance Management – Realizing APM”.