Automatic Problem Detection with Dynatrace

An excerpt from the Dynatrace blog

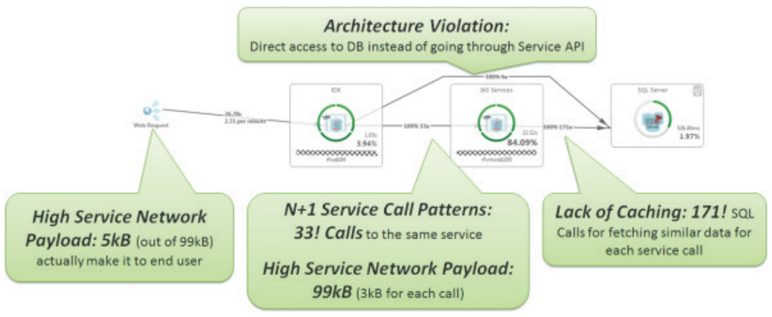
Best Practices

About the Author

Andreas Grabner has been helping companies improve their application performance for 15+ years. He is a regular contributor within Web Performance and DevOps communities and a prolific speaker at user groups and conferences around the world. Reach him at @grabnerandi

For the complete version of this blog, go to: http://bit.ly/2pe6CNG

Can you imagine automatic problem detection being a reality?! What would it take to make it possible, practical and functional? Over the years we at Dynatrace have seen a lot of PurePaths being captured in small to very large applications showing why new deployments simply fail to deliver the expected user experience, scalability or performance. Since I started my Share Your PurePath Program a little over a year ago I received about 250 Dynatrace Sessions from our mainly free trial / personal license users. My free performance analysis not only helped our users to solve their application performance issues, I also gathered many great stories to tell at conferences, my Online Performance Clinics on YouTube, our PurePerformance Podcast or through blog posts such as Top Tomcat Performance Problems, C# Performance Mistakes, PHP Performance Hotspots or more!


Architecture Violation:
Direct access to DB instead of going through Service API
Web Request
IDK
IAS Services
SQL Server
3.94%
84.09%
1.97%
High Service Network Payload: 5kB (out of 99kB) actually make it to end user
N+1 Service Call Patterns: 33! Calls to the same service
High Service Network Payload: 99kB (3kB for each call)
Lack of Caching: 171! SQL Calls for fetching similar data for each service call

One of my recent highlights I keep referring to: Monolith to Micro-Service Gone Bad!!

Common Problem Patterns around the Globe

The shocking truth is that most applications suffer from the same performance, scalability and architectural issues. The good news is that we can easily detect them by looking at the right metrics that Dynatrace captures. Instead of having me look at your PurePaths all the time or having you spend time to Drill Down here and Drill Down there our engineering team sat down with me to learn which problem patterns I typically go after and how I do it.

We came up with problem pattern categories and buckets for each category that I use to "tag" a PurePath saying that the PurePath suffers from a certain problem:

Response Time

Buckets: Very Fast (<100ms); Fast (100 – 500ms); Normal, (500 – 1000ms); Slow (1-4s); Very Slow (> 4s)

My first analysis looks at speed of PurePaths. I want to analyze very slow once but also compare them with the same requests on the same URL that went really fast. In that case it comes handy that Dynatrace captures all transactions all the time – not as other APM tools which only capture outliers or start sampling. In my case I can spot the slow ones but also compare them method by method with the fast ones.

Complexity

Buckets: Complex (> 10 Agents or > 1000 Nodes); Medium (< 10 Agents or < 1000 Nodes); Normal (< 3 Agents or < 100 Nodes)

I see many people sort PurePaths by Size (=Number of Nodes) as these are typically the more complex transactions with more things that can go wrong. Well – I do the same thing. Additionally to the nodes I also look at the number of tiers involved (=Number of Agents). If you have a transaction that spawns more than 10 Tiers then there is a highly complex transactions with lots of potential problems!

Threading

Buckets: Heavy (> 5 Threads per Path); Acceptable (< 5 Threads per Path); Normal (<=2 Threads per Path)

I've seen a lot of transactions that make heavy use of background threads to parallelize work. Its not a bad thing but most of the time I see a lot of bad coding and bad synchronization. Also – the more threads involved the more resources you consume on the App Server. The more complex the system becomes which is why I always look at the total number of threads involved in processing a single transaction

Asynchronous

Buckets: Heavy (Duration > 10x of Response Time); Normal (has async activity); No Async

Response Time is good to optimize but you also need to look at transactions that respond fast but then run in the background forever. For that reason Dynatrace also captures PurePath Duration which includes all asynchronous activity triggered by the initial request. If I see an unusual long Duration as compared to the Response Time it typically means that something is wrong.

Content Type

Buckets: Static (requests to images, css, javascript, html, …), Dynamic

I most often filter out static requests when I try to analyze server side performance problems as static request are handled by the web server. To filter out the noise I simply just focus on dynamic requests. On the other side it is interesting to see how many static requests make it to the app server. If the number never declines it means we do not really make use of client side caching!

Database

Buckets: DB Chatty (> 100 SQL Executions); Single Long SQL (1 SQL taking > 80% of SQL Total); N+1 Query Problem (Same SQL executed > 10 times); Overall DB Time High (JDBC/ADO.NET/PDO time > 80% of Total Exec Time)

Database access is the #1 problem pattern we see in apps. This is why we analyze database access and mark PurePaths that show strange patterns. Even if response time of the application is fast we want to know whether our developers added a pattern that will cause problems later on when we have more traffic on the system or a larger database. This type of analysis also only works if you always capture every single transaction with all details which is what Dynatrace does but other APM vendors don't! You also need this data to compare the same business transaction that shows a bad access pattern vs one that does not. Classic example is a bad search query that causes lots of database queries. For that to know you need to have the good and the bad transaction available and compare!

Performance Breakdown

Buckets: CPU (1 Method consumes > 80% overall CPU); Wait (>= 50% Wait Time); Sync (>= 50% Sync Time)

Once I find a slow transaction I first look at whether it has a CPU, Wait or Sync hotspots. This is why it makes perfect sense to put PurePaths into these buckets. I can immediately focus my analysis on PurePaths that block each other out via synchronization blocks. I can also identify different transactions that wait on a shared resource!

Web Service

Buckets: High payload (Request + Response Size > 500k); Chatty (> 50 Service Calls per Request); Slow (1 Service Call > 80% of Total WS Time); N+1 Pattern (same Service call more than once); High Web Service Time (> 80% Total Time)

Services have always been used by applications. With the gained popularity of "MicroServices" and lots of frameworks that make it very easy to write and consume services we see a lot of bad web service access patterns. Like in the screenshot at the very top: a PurePath that shows 33 Web Service Calls where most of them are the same and that consume > 80% execution time is a problematic PurePath

Log Activity

Buckets: Spamming (> 50 Messages); High (> 10 Messages); Normal (< 10 Messages); No log

Dynatrace captures log messages as part of the PurePath which makes it perfect to identify critical problems but also log spamming per transaction. Even though we have Splunk and ElasticSearch it doesn't mean we should log too much data that gives no value. Too much logging in that case can become a performance problem as well as impacts I/O and potentially the costs you pay to your log analytics tool.

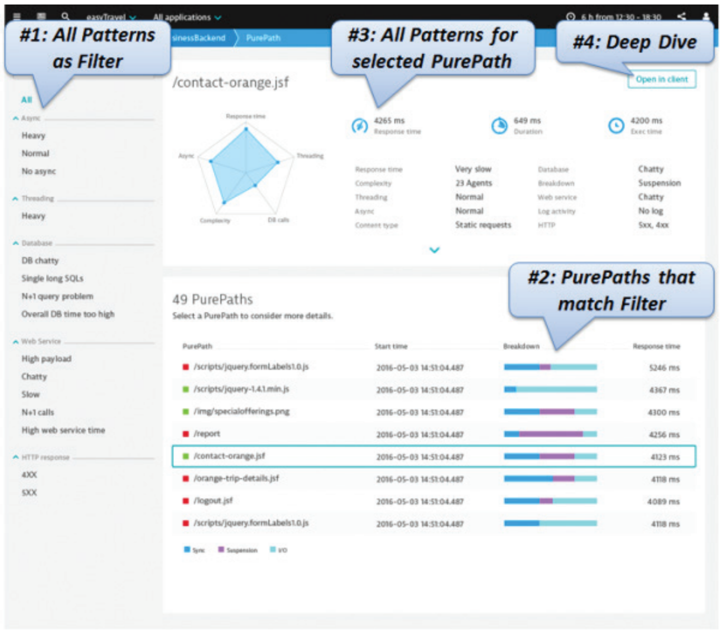
HTTP Response Codes

Buckets: 2xx, 3xx, 4xx, 5xx

I often start with looking at requests that return a certain HTTP Status Code. It is very interesting to analyze failing requests that return an HTTP 5xx. It is also interesting to learn about redirects or authentication issues. This is why I categorize PurePaths based on HTTP Status Codes

Automatic Problem Pattern Detection

I hope you learned a thing or two on how I go about analyzing PurePaths. But I kept the good news until the end! Our engineering team actually automated the whole detection process for us. Really! Starting with Dynatrace 6.5 (EAP and Free Trial to start July) every PurePath will be analyzed and tagged with one or more of the Problem Patterns. A new PurePath view in our Web UI then allows us to find PurePaths with a particular pattern and shows us what other patterns this PurePath suffers from. My colleague Alois recently named this "Andi in a Box"!



/contact-orange.jsf

4265 ms Response time — 649 ms Duration — 4200 ms Exec time

Response time	Very slow	Database	Chatty
Complexity	23 Agents	Breakdown	Suspension
Threading	Normal	Web service	Chatty
Async	Normal	Log activity	No log
Content type	Static requests	HTTP	5xx, 4xx

49 PurePaths
Select a PurePath to consider more details.

PurePath	Start time	Breakdown	Response time
🟥 /scripts/jquery.formLabels1.0.js	2016-05-03 14:51:04.487		5246 ms
🟩 /scripts/jquery-1.4.1.min.js	2016-05-03 14:51:04.487		4367 ms
🟩 /img/specialofferings.png	2016-05-03 14:51:04.487		4300 ms
🟥 /report	2016-05-03 14:51:04.487		4256 ms
🟩 /contact-orange.jsf	2016-05-03 14:51:04.487		4123 ms
🟥 /orange-trip-details.jsf	2016-05-03 14:51:04.487		4118 ms
🟥 /logout.jsf	2016-05-03 14:51:04.487		4089 ms
🟥 /scripts/jquery.formLabels1.0.js	2016-05-03 14:51:04.487		4118 ms

Sync Suspension I/O

"Andi in a Box": Automatic Problem Pattern Detection per PurePath in Dynatrace Web Dashboards

This will make all our lives so much easier as the detection of 80% of the problems we found in the recent years are now fully automated. This gives us time back to actually analyzing the tough problems that we haven't yet had time to examine. Or spend time with other projects, friends and family!

Give it a try

Everyone can give this a try. Simply register for the Dynatrace Free Trial and install it on your app. Once you capture PurePaths open your Web Dashboard (https://localhost:9911) and through the "burger menu" click on PurePaths.

The rules that we apply now come with default settings but can be changed. So if you don't like some of my recommendations simply apply yours. But please let us know as we want to learn and adapt to better meet your needs. Simply post comments on https://answers.dynatrace.com

Dynatrace Digital Performance Platform — its digital business...transformed.

Successfully improve your user experiences, launch new initiatives with confidence, reduce operational complexity and go to market faster than your competition. With the world's most complete, powerful and flexible digital performance platform for today's digital enterprises, Dynatrace has you covered.

04.25.17 2236_FS_jw