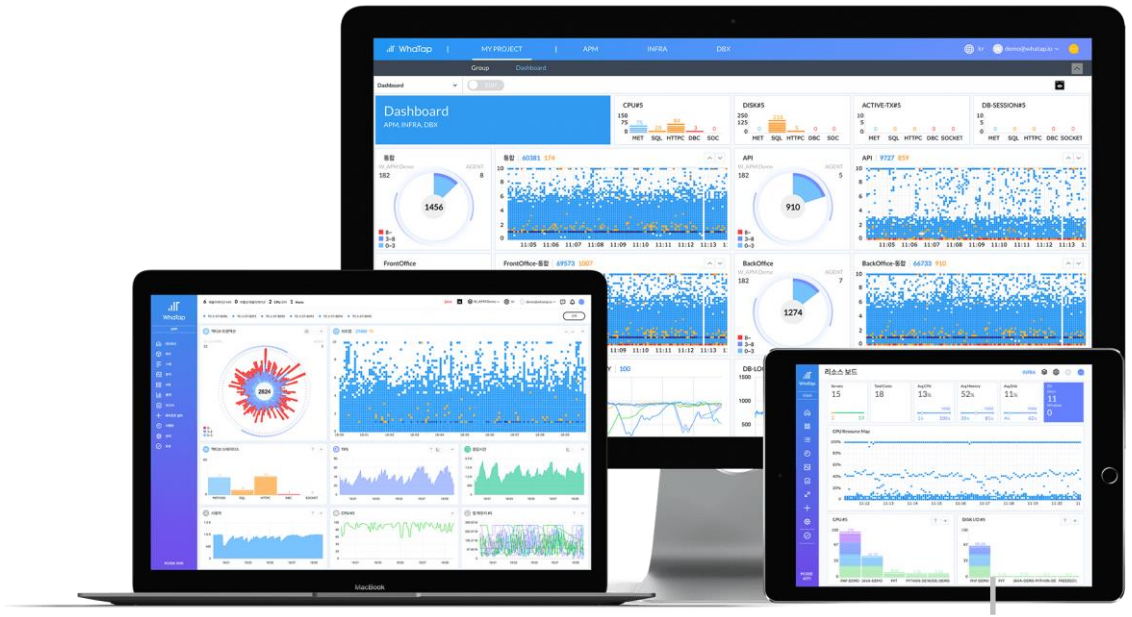
WhaTap

WhaTap Monitoring Service

Case Studies



WhaTap Labs

당신의,
IT 서비스의 성능을 극대화합니다.

Contents

WhaTap Labs

IT Monitoring Service based on SaaS.

Provide monitoring service optimized for cloud.

Maximize the performance of IT service.

Hidden formatting causing CPU issue was found.

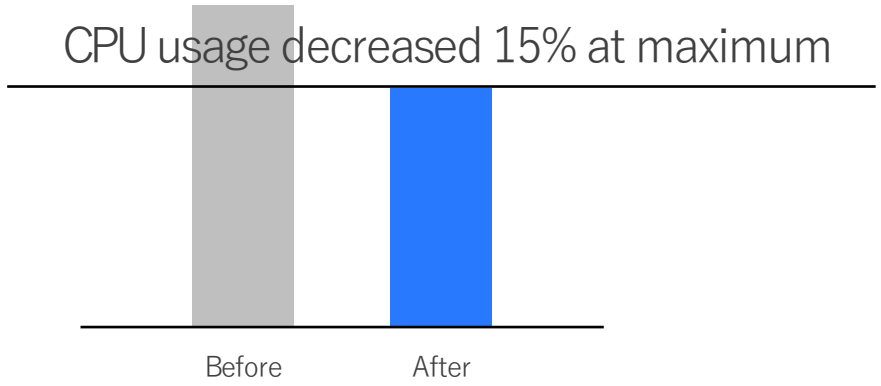
Background

- When purchasing smart phone is heavily increasing on weekend or after holiday, application's CPU usage is also increasing. To overcome this, the operators are restarting the applications.
- After iphone 7 was launched and users were starting to buy the phone, the application couldn't overcome this and be downed.

WhaTap Analysis Found the common module has been using specific method excessively.

Results

- CPU usage decreased about **10~15%**
- Application got to be stable
- LG Uplus started to use WhaTap on entire site



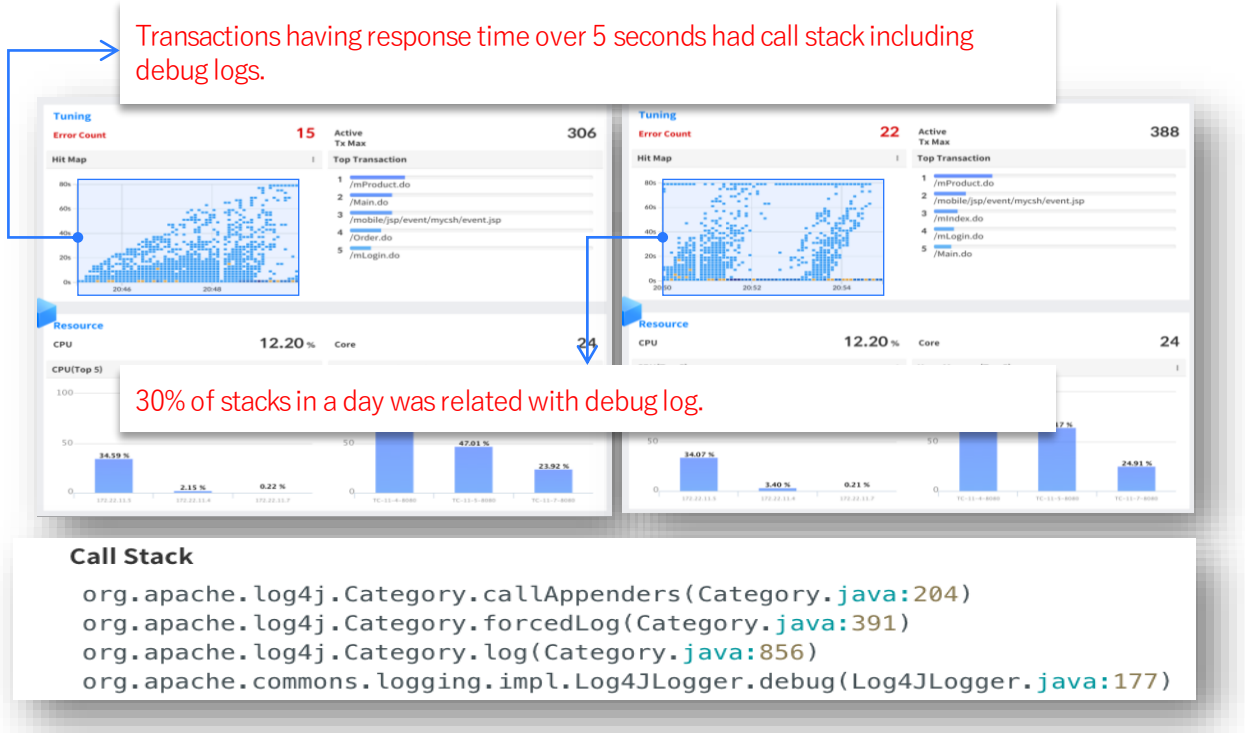
Application latency caused by debugging logs

AMOREPACIFIC

Background

- Application latency on Single's Day in China happened.
- There were too many logs and couldn't estimate the relation between logs and performance loss.

WhaTap Analysis Majority of stack was found on debug logs.



Results

Operation team turned off the debug log.

The latency rate of transaction went down.

Before

퍼센트	건수	
60.02%	52885	java.net.SocketInputStream.socketF
31.35%	27618	org.apache.log4j.Category.callAppe
5.89%	5192	sun.misc.Unsafe.park(Native Metho

After

퍼센트	건수	
72.07%	252816	java.security.SecureRandom.nextBytes(Secu
12.58%	**44148**	**sun.misc.Unsafe.park(Native Method)**

(99.99%) java.util.concurrent.locks.Loc

WhaTap

Cooperation with experts via SaaS

ASIANA AIRLINES

Background

- PoC on a single server with a single scenario on Asiana Hompage

WhaTap Analysis

Lack of connection

Slow paging SQLs

Single Sign On latency

Needless duplicate call of sub pages

Results

- Experts provided the reports multiple times and got to be trusted.
  - Sizing up the connection pool
  - Advised to cache the menu caching structure
  - Advised to change the call structure between main page and sub pages.
- We maximized the merits of cooperation via SaaS on this case.

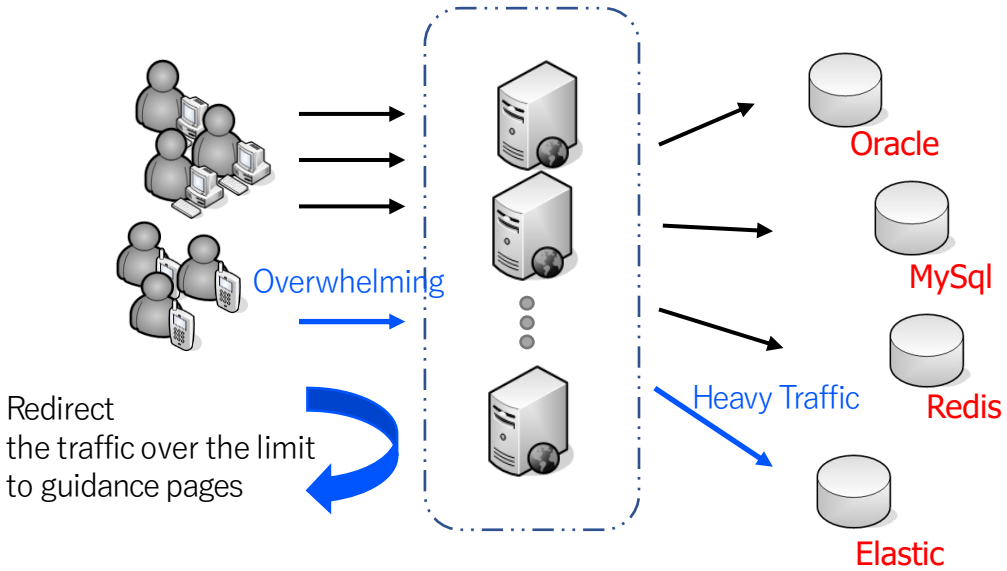
WhaTap

Prevent the service hanging by throttling

GS SHOP

Background

- Steep user increase caused the site hanging whey they were going on promotion.
- They needed to block heavy traffic to flow onto backend system.

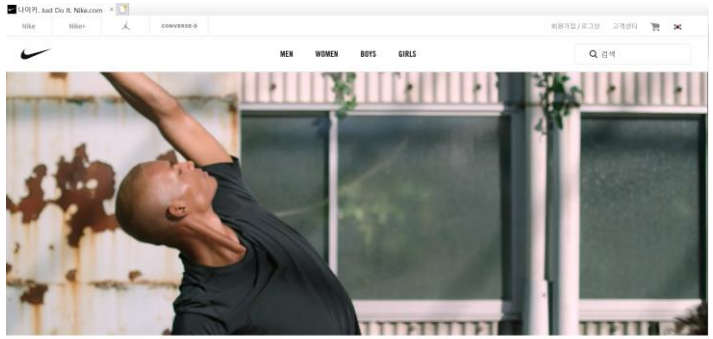


Heavy traffic over the limit of backend
make scale-out impossible.

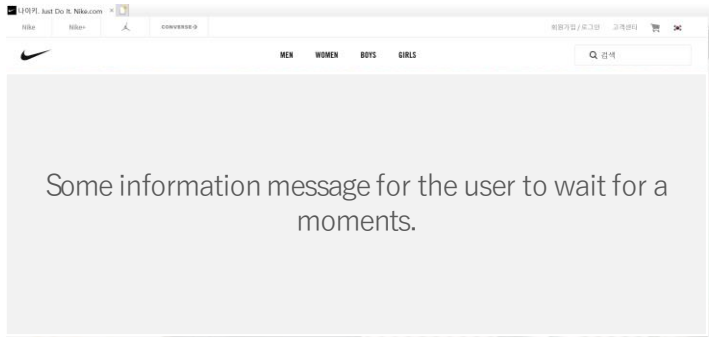
Results

- Depending on active transaction count,
  forward or redirect the transaction to static pages to control the entire traffic.
- Controlled just some specific event pages by throttling.

At normal status



Guidance page at heavy traffice



WhaTap

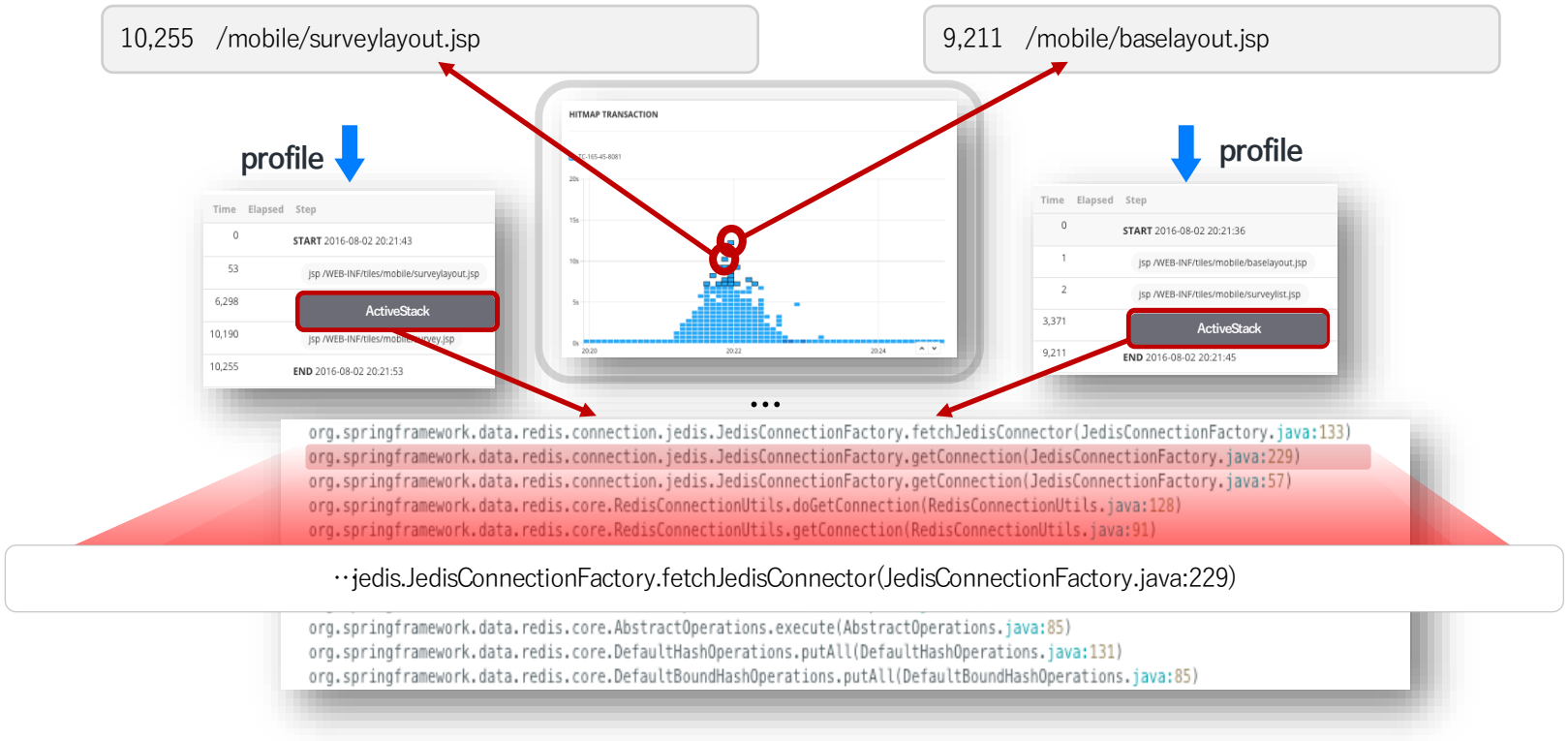
Bottleneck of Redis

Background

- When launching global system, it get harder to detect the reason of latency.
- They didn't have enough man power to monitor and analyze all the system.

Results

- Active Stack reduced the overhead to the operators and make it easy to analyze and detect core reason of latency.



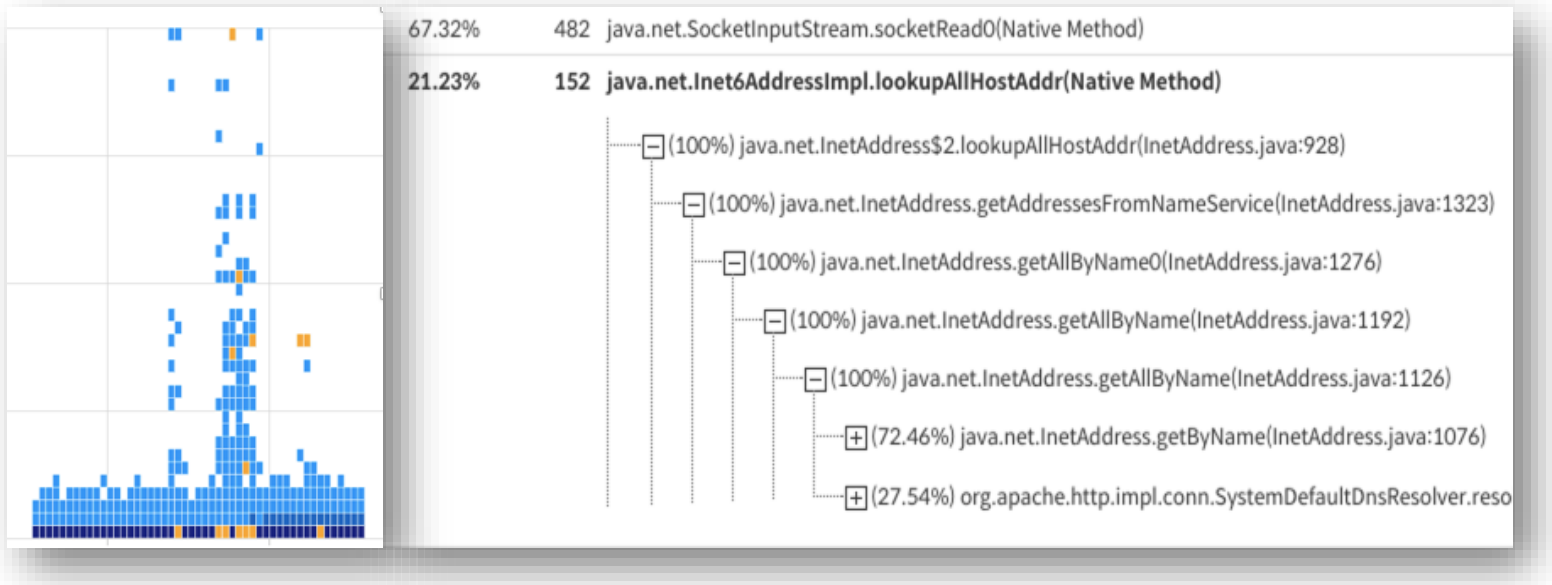
Bottleneck of Host lookup

Background

- Service latency happened, but they were unable to estimate the reason and were suspicious about the network.

Results

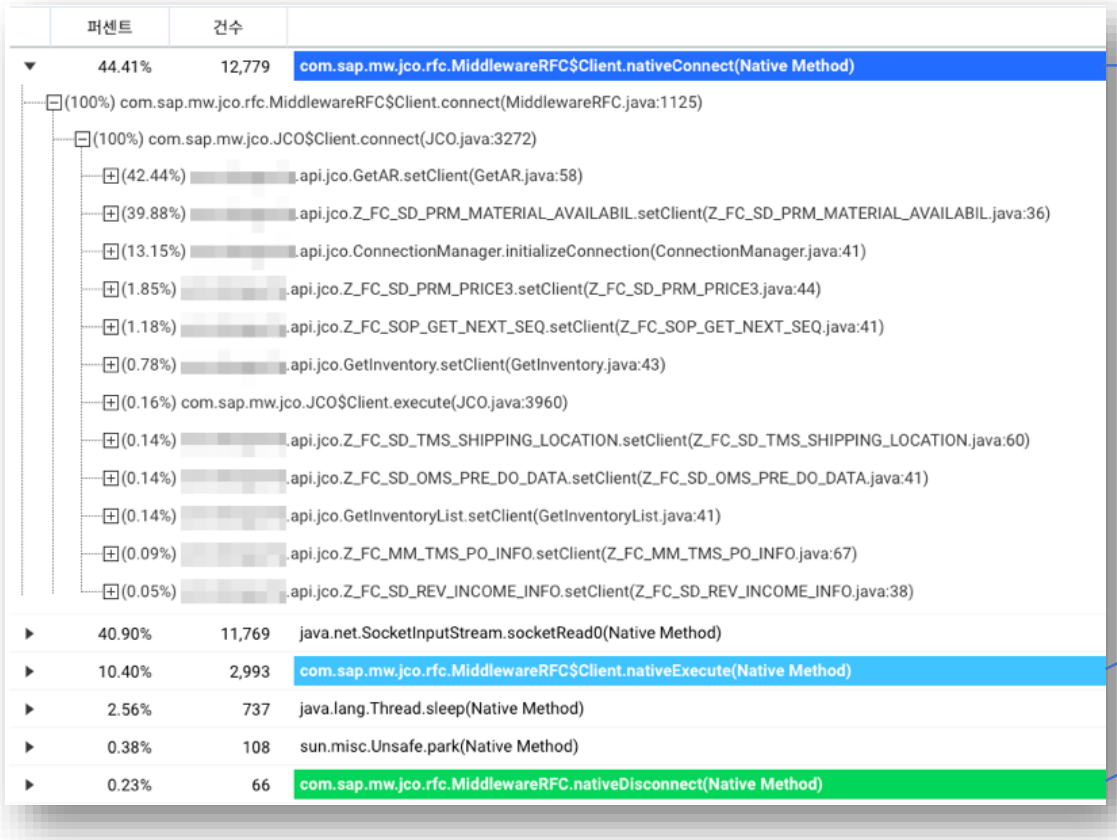
- IPv6 Address Lookup was detected by stack statistics.
- Operators succeeded to solve the problem by adding the JVM option. -Djava.net.preferIpv4Stack=true
- Faster the average response time of transaction about 20%. 120ms => 97ms



Bottleneck of SAP ABAP

Results

- When monitoring SAP, we have detected the latency of native connection of ABAP jco library.
- We advised **to use connection pool** for improving this.



Connection Overhead 44.41%

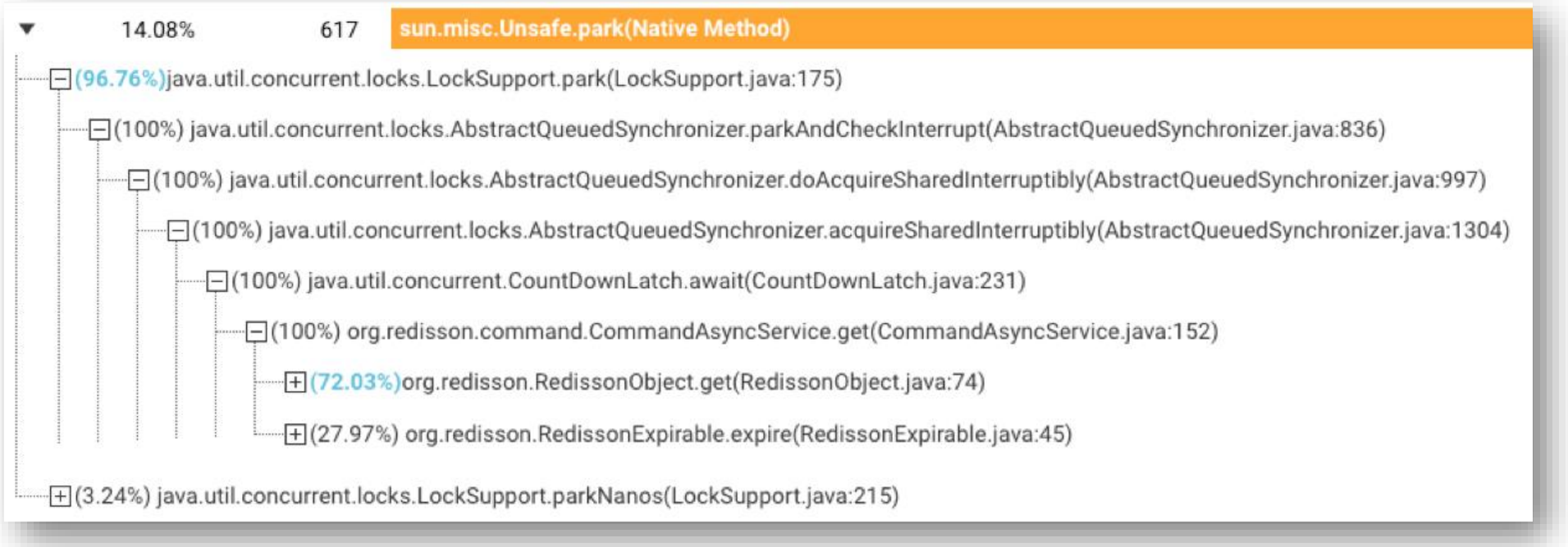
ABAP running 10.4%

Close connection 0.23%

Bottleneck of Reddison

Results

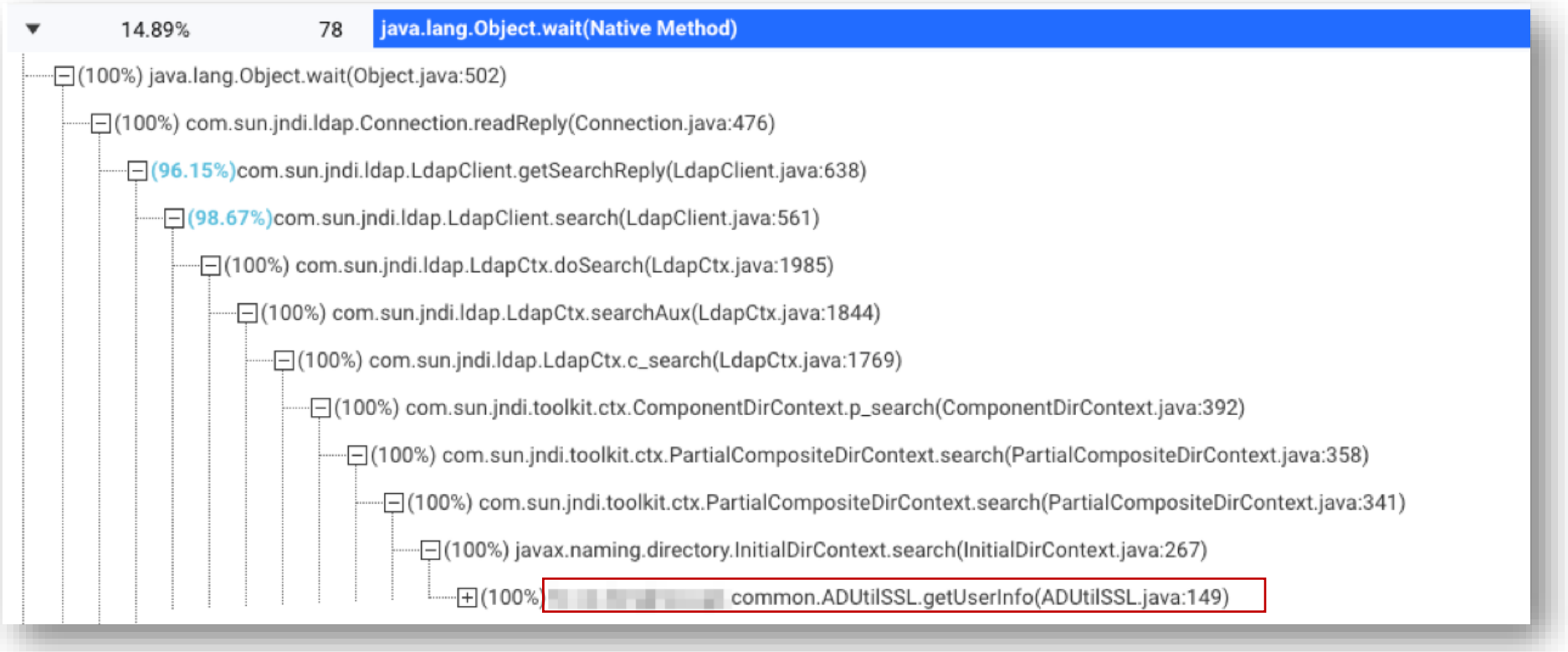
- Reddison is the library for sharing. When monitoring applications using this library, the bugs of this library caused the latency of entire system.
- We advised the performance improvement will be about 14%, if the customer can solve the bottleneck.
  https://github.com/redisson/redisson/releases



Bottleneck of ActiveDirectory & LDAP

Results

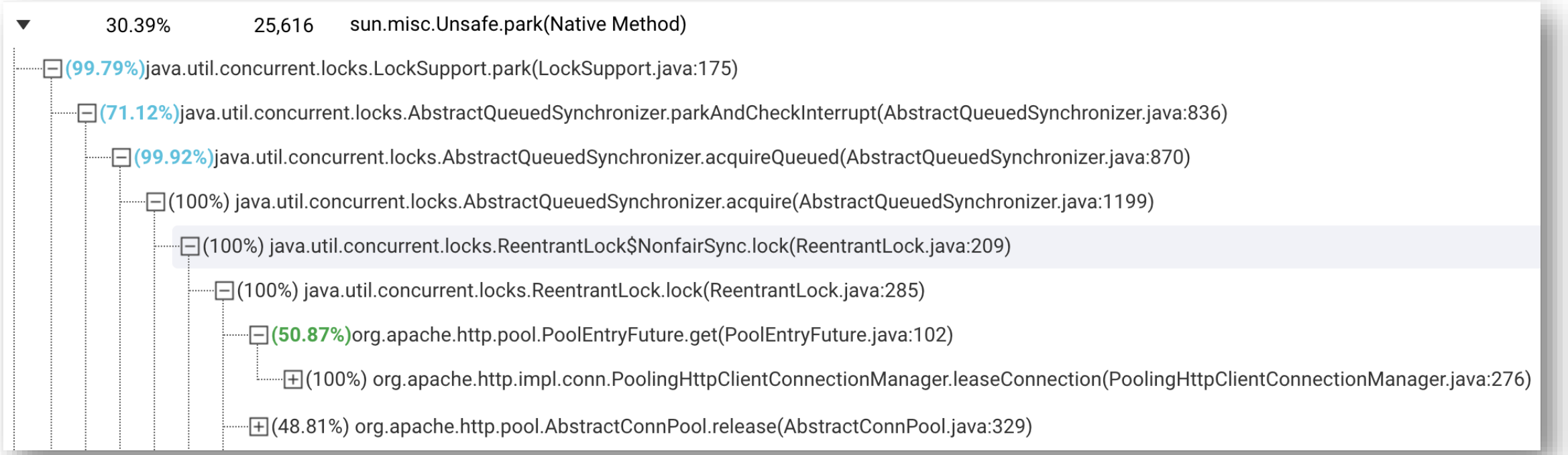
- When monitoring applications using Active Directory and LDAP, major stack was found on the calls related with getting user information from LDAP.
- We advised to **use TLS over 389**.
  https://docs.oracle.com/javase/jndi/tutorial/ldap/ext/starttls.html



Bottleneck of lease and release of HTTP Client pool

Results

- When monitoring applications using connection pool, we detected bottleneck in the process of leasing and releasing connection from pool.
- We advise to expand the pool size like this case.

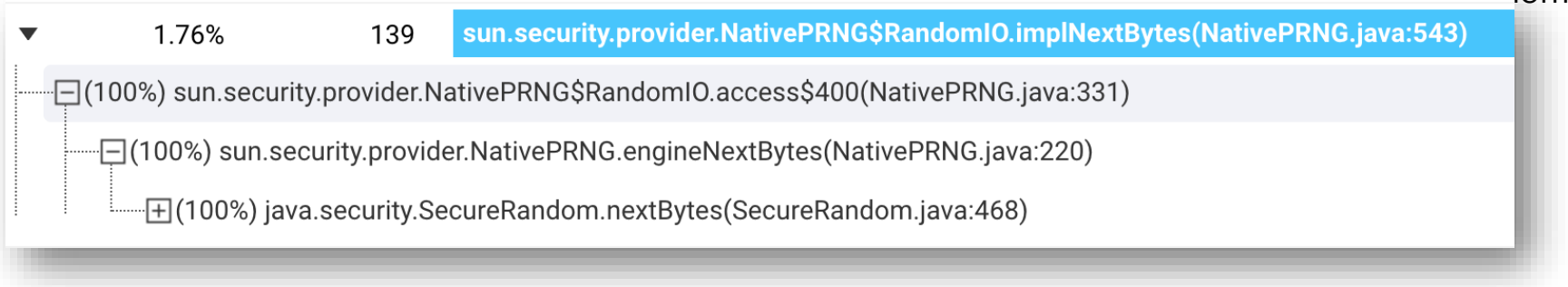


Bottleneck of NativePRNG

Results

- Overhead made from encryption and decryption can be bigger as the TPS goes high.
- We advised to replace the random seed of encryption and decryption because to improve the performance of it makes greater gain of entire performance.

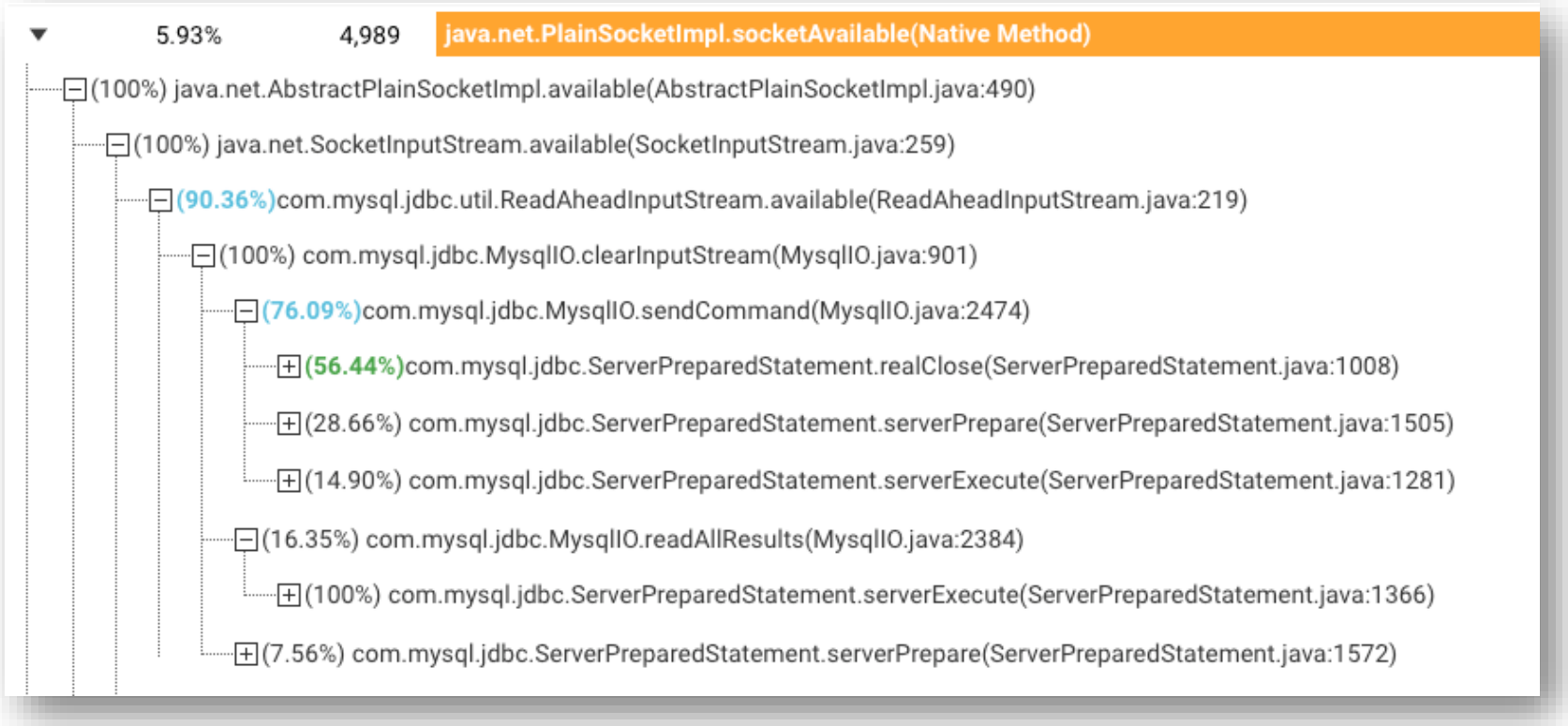
-Djava.security.egd=file:/dev/./urandom => Performance gain above 1% comparing to /dev/random



Bottleneck of JDBC Driver

Results

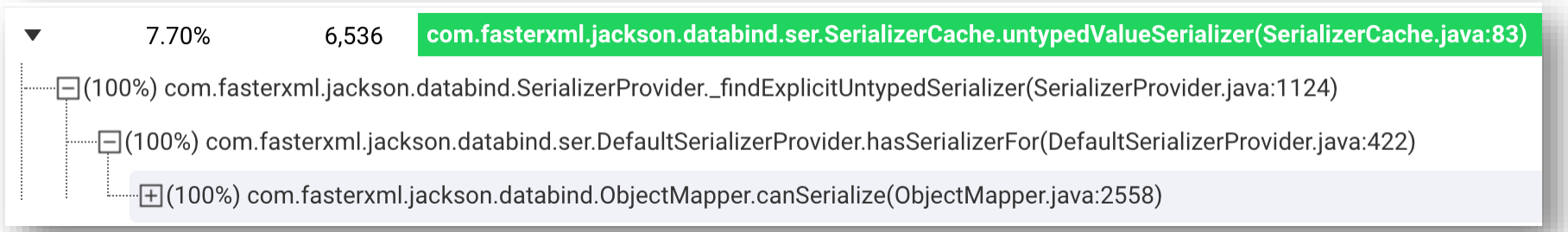
- For the very high TPS applications, blocking on JDBC driver must not be found.
- We advised to update the jdbc library because the version used by customer has a bug.
  https://dev.mysql.com/doc/relnotes/connector-j/5.1/en/news-5-1-43.html



Bottleneck of JSON Serialization

Results

- For the very high TPS applications, to make serialization faster is very important.
- We advised to update the fasterxml library because the version used by customer has a bug having blocking issue.
  https://github.com/FasterXML/jackson-databind/issues/1180



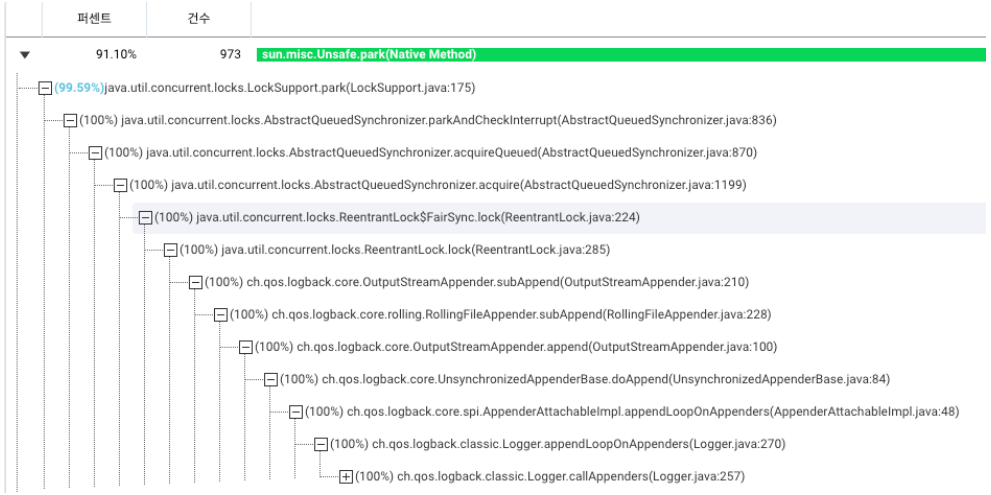
Race condition of logging

Results

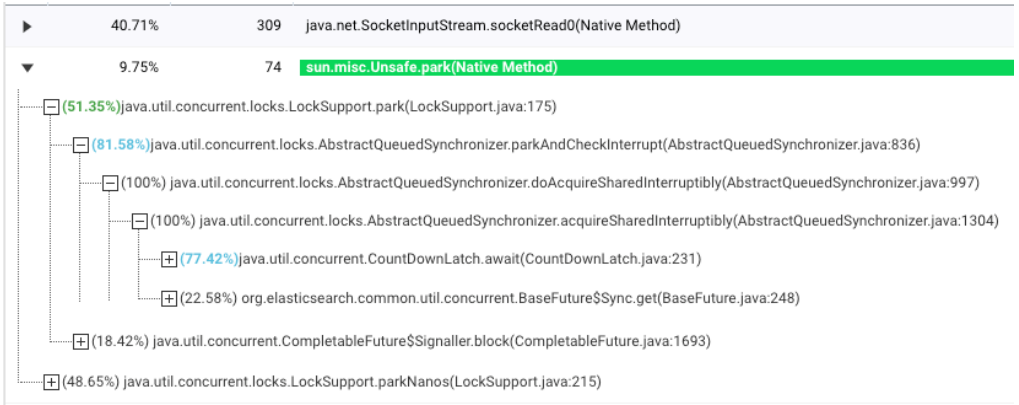
- The new promotion event caused a performance issue due to a spike in usage.
- This is due to contention in the logback module, which can be solved by reducing the amount of logs and update logback library.
  https://docs.google.com/spreadsheets/d/1cpb5D7qnyye4W0RTlHUnXedYK98catNZytYIu5D91m0/edit#gid=0

WhaTap Analysis Found the logging module's race condition.

java.util.concurrent.locks.ReentrantLock$FairSync.lock (ReentrantLock.java:224)



Results After reducing the amount of logs, the problem was resolved.



More with WhaTap Monitoring Service

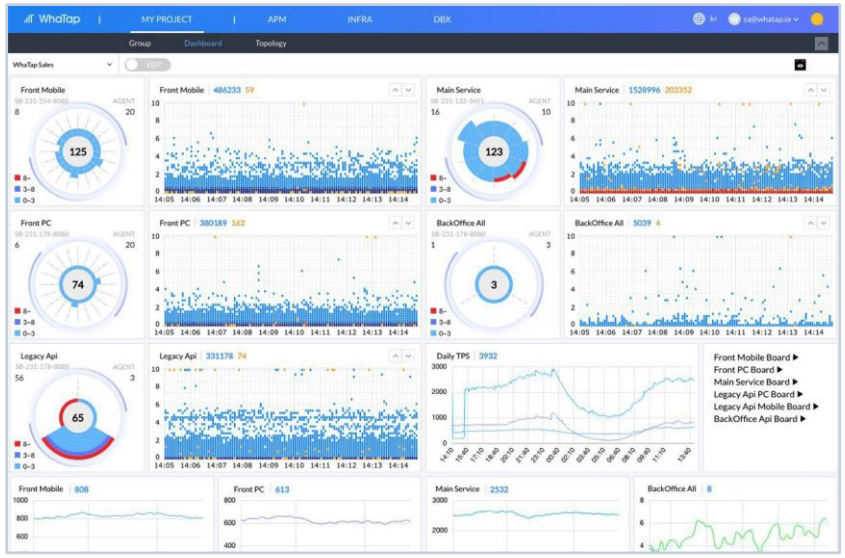
롯데면세점

Background

- They were already using global solution, but cannot use it fully.
- They need some experts to service the analysis.

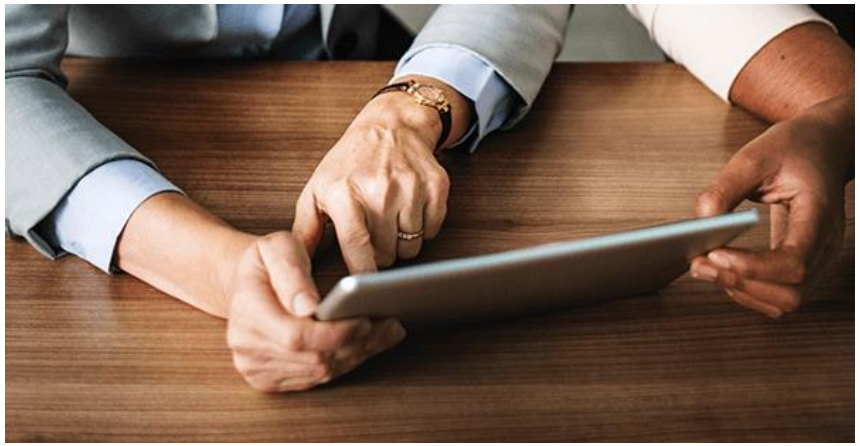
WhaTap Analysis

- By using WhaTap Monitoring Service, they could be supported by WhaTap engineer.
- When application latency happens, customer and WhaTap engineer cooperate on it.



Results

Cooperation can be more than just using a solution.



WhaTap

Thanks