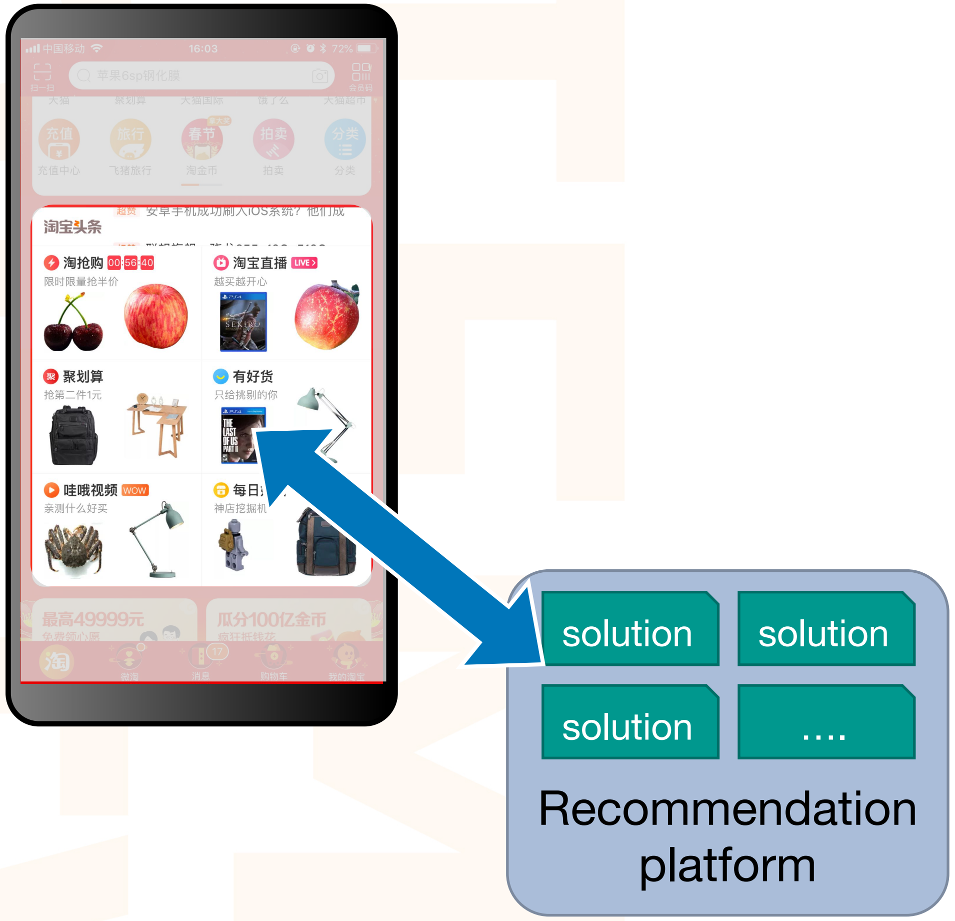
FaaS and Dynamic AOT

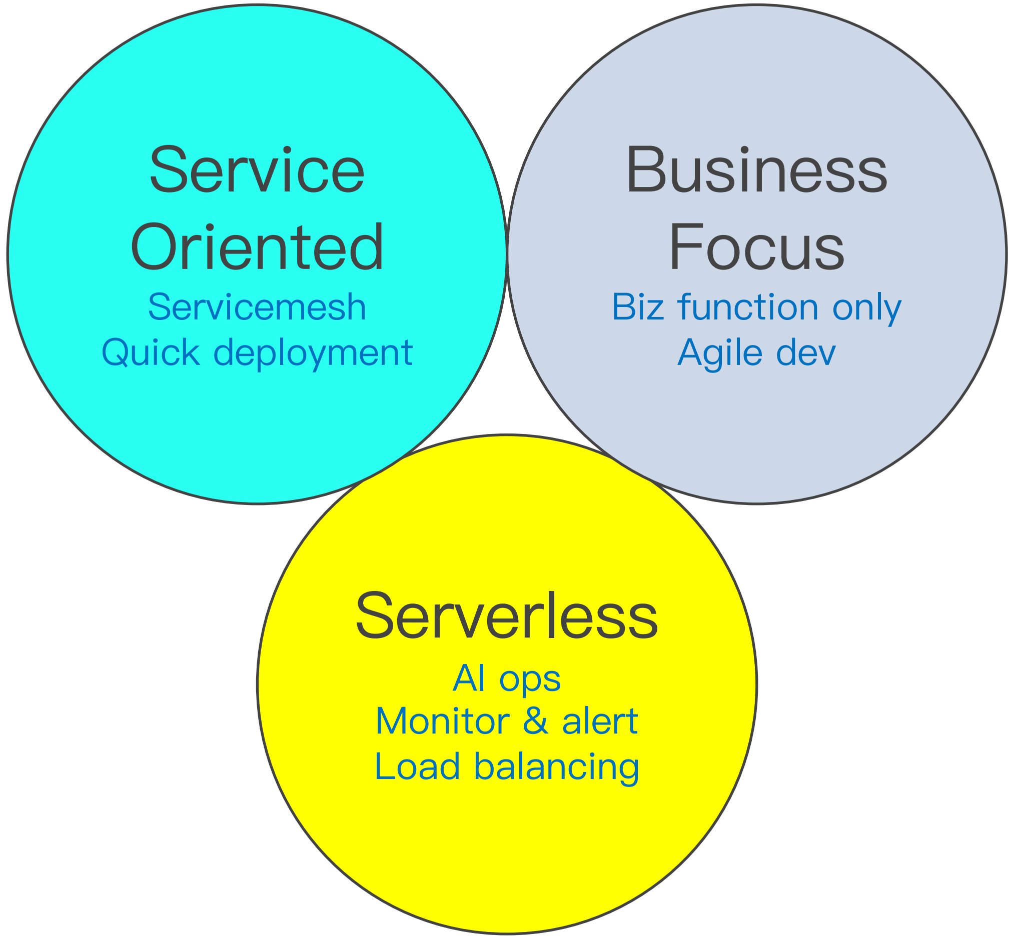
Kuai Wei, Zhang Yifei, Wang Zhuo, Li Sanhong, Lu Chuansheng

Alibaba System Software

FaaS-based recommendation platform

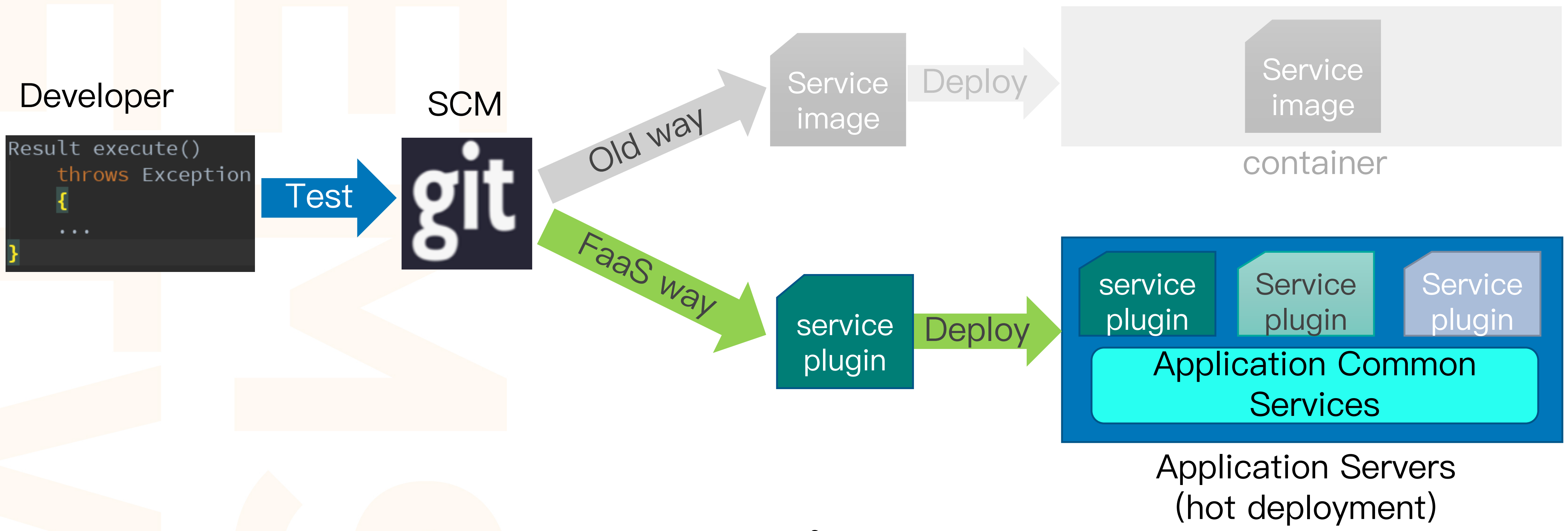
Alibaba System Software





FaaS style release model

- Hot deployment based
- Accelerate application deployment:
  - Week → hours or minutes



```
Result execute()
    throws Exception
    {
    ...
}
```

Challenges to JVM

- Service startup time is critical
  - Resource is allocated on demand
  - WarmUp (interpreter + JIT) time must be reduced
  - Move some work before full JVM startup
- Footprint should be controlled
  - CPU/Memory resource is limited
  - App server may return resource in idle time
- No performance regression
  - For running phase

Our current choice: OpenJDK AOT

- OpenJDK AOT vs Graal native–image
  - Both are ahead–of–time compilation
- Graal native–image usually has
  - smaller footprint
  - quicker startup time

- But Graal native–image
  - needs close–world compilation
    - All java classes must be known to compiler
  - cannot go back to JIT mode
    - For dynamic generated classes/redefined class
    - AOT can go back to normal execution (interpreter/JIT)
  - does not support hot deployment
    - AOT does not support it either
    - But AOT could be enhanced to support it

Dynamic AOT in AJDK

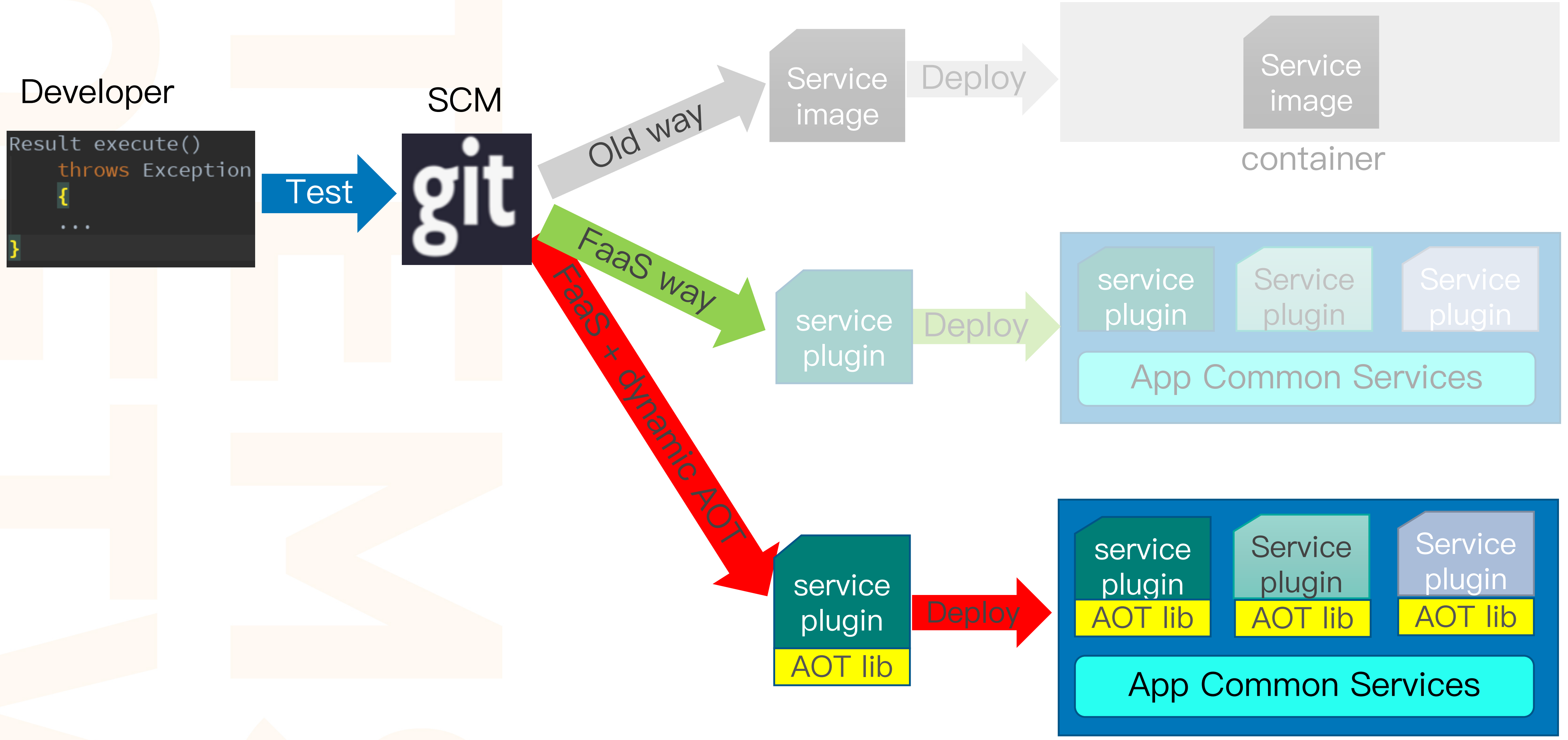
- AOT library is bound to corresponding class loader
  - *Service plugin* is always loaded by a separate class loader
- Java APIs to load/unload AOT libraries

```
public static synchronized int loadAOTLibraryForLoader(ClassLoader loader, String library)
```

```
public static synchronized void unloadAOTLibraryForLoader(ClassLoader loader)
```

- Same AOT library can be loaded for multiple times by different loaders

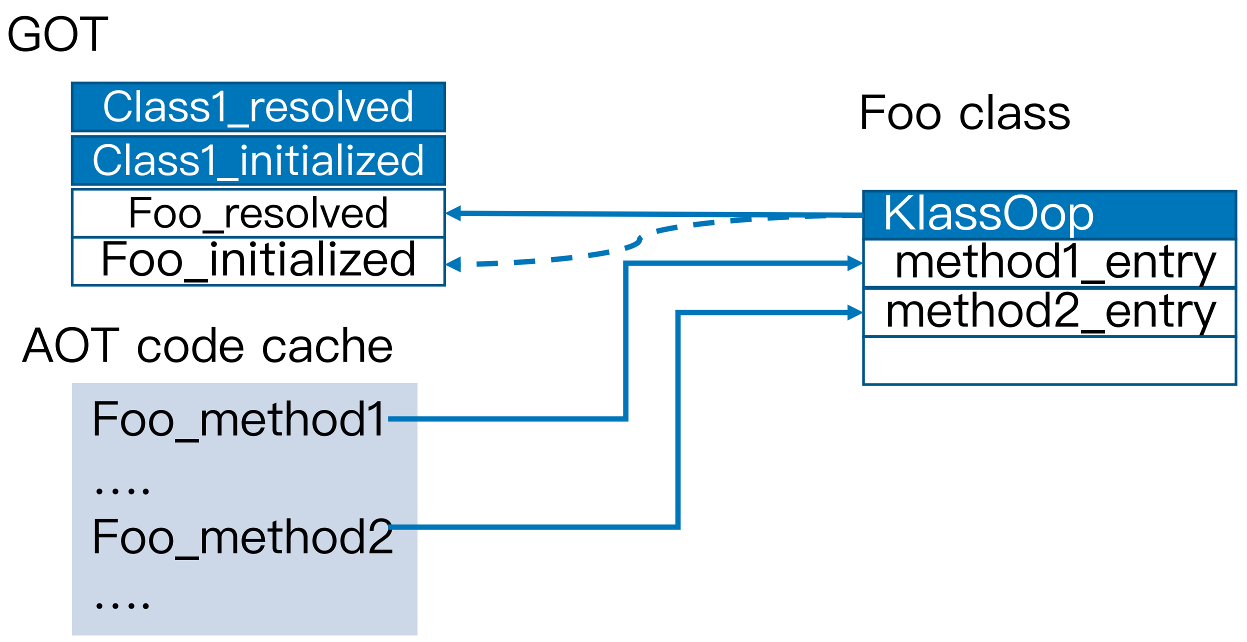
FaaS + Dynamic AOT



```
Result execute()
    throws Exception
    {
    ...
}
```

Class resolution in Dynamic AOT

- GOT(Global Offset Table) is used to store resolved *klassOop*
- When a class is loaded, JVM checks AOT code cache to find referenced class
- Assign native methods in AOT cache to resolved class
- AOT generated code will check GOT to know if referenced class is resolved or not
- For dynamic AOT, every referenced class should be checked because class may be loaded before loading library

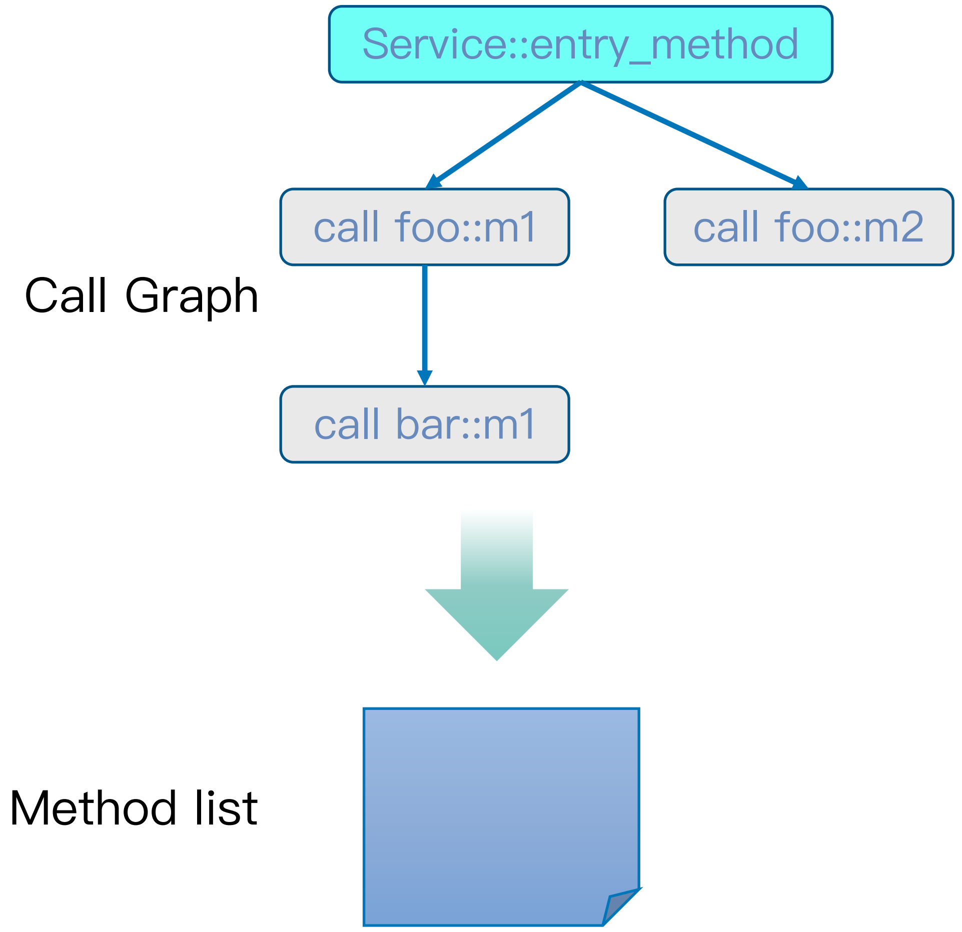


AOT code

```
mov     0x20fd81(%rip),%rax          # 2120a0 <got.init.Foo;>
test    %rax,%rax
je      2349 <Foo.add(II)I+0xa9>     # call init stub
```

Enhance Dynamic AOT by using static analysis

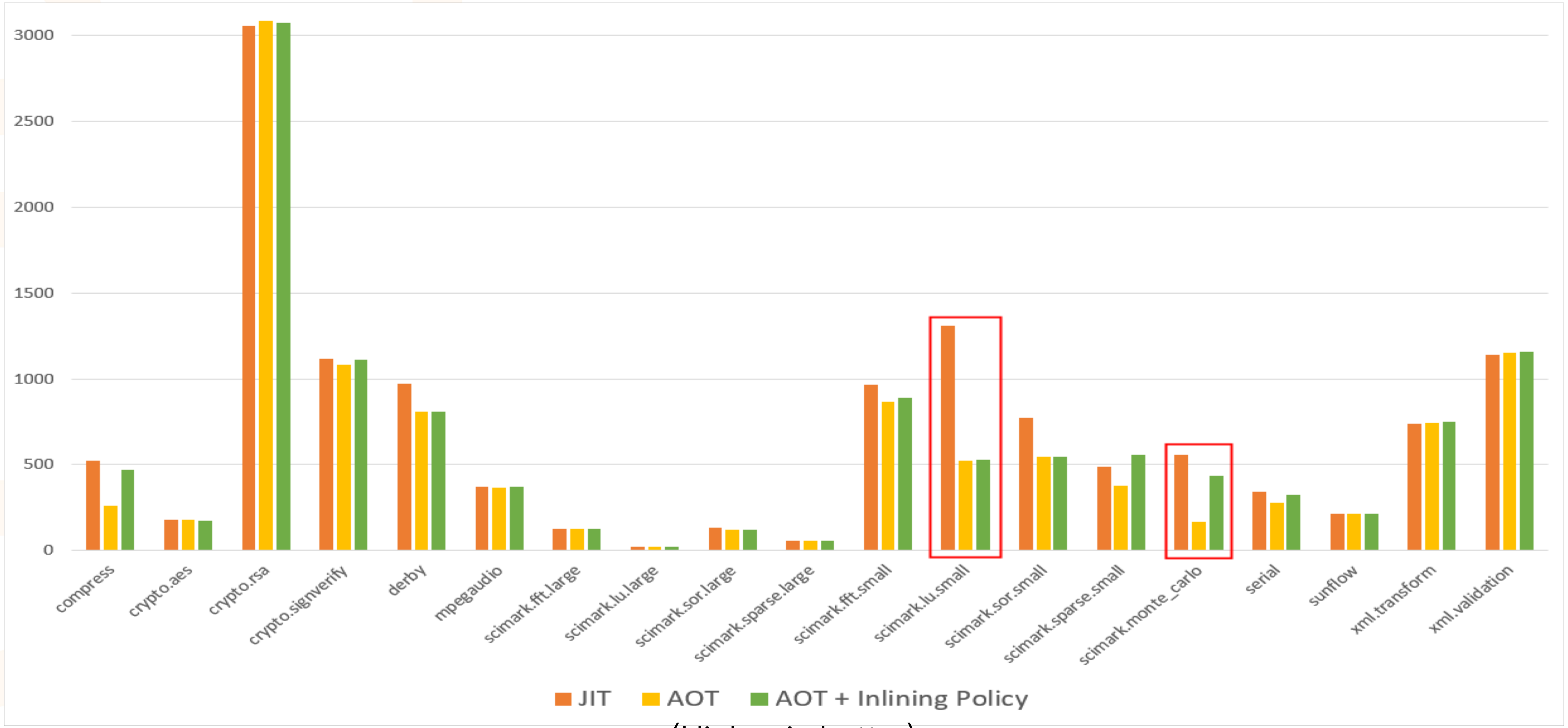
- Construct call graph from entry methods of service
  - SOOT framework is used
- Use call graph to generate compilation method list
- Just cover those mostly used methods
  - Size is reduced to 50%
  - Cover 90% of used methods



Enhance AOT with inline change

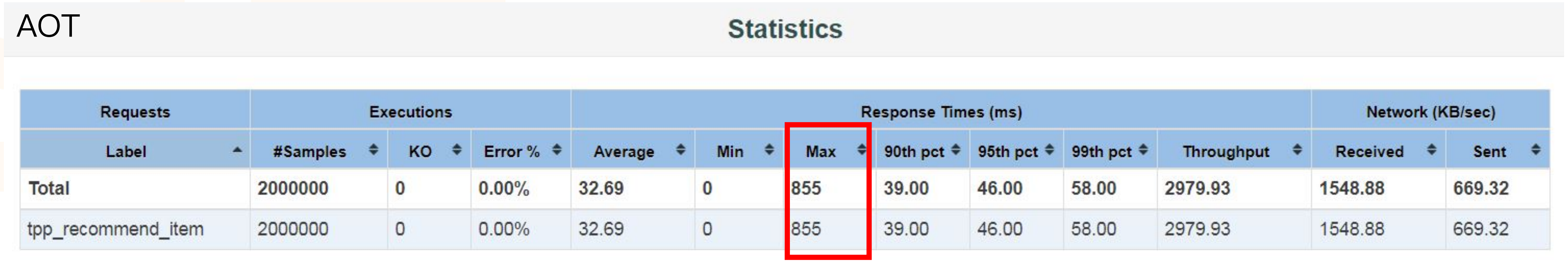
- When jaotc compiles methods
  - if a class has not been initialized, current impl will **NOT** inline it.
- Make jaotc to load/initialize class on demand and inline it
- Will improvement SPECjvm.monte_carlo a lot

AOT Performance — SPECjvm2008



(Higher is better)

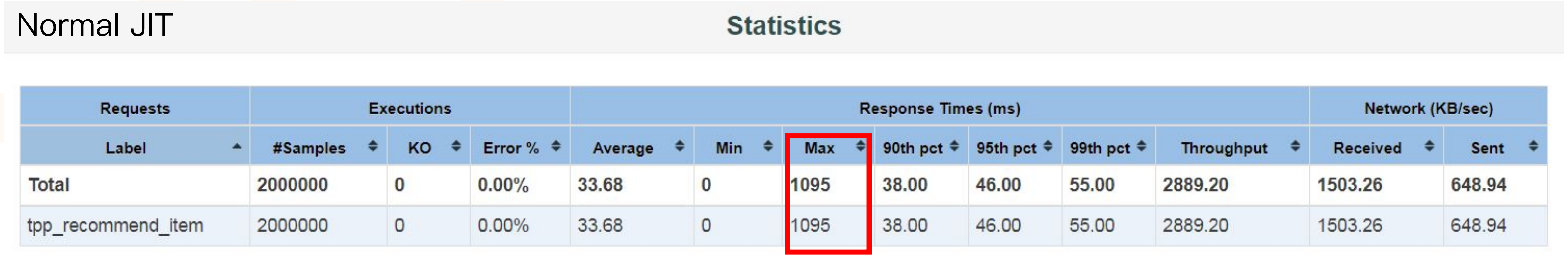
AOT performance – response time



AOT

Statistics

Requests	Executions			Response Times (ms)						Network (KB/sec)		
Label	#Samples	KO	Error %	Average	Min	Max	90th pct	95th pct	99th pct	Throughput	Received	Sent
Total	**2000000**	**0**	**0.00%**	**32.69**	**0**	**855**	**39.00**	**46.00**	**58.00**	**2979.93**	**1548.88**	**669.32**
tpp_recommend_item	2000000	0	0.00%	32.69	0	855	39.00	46.00	58.00	2979.93	1548.88	669.32

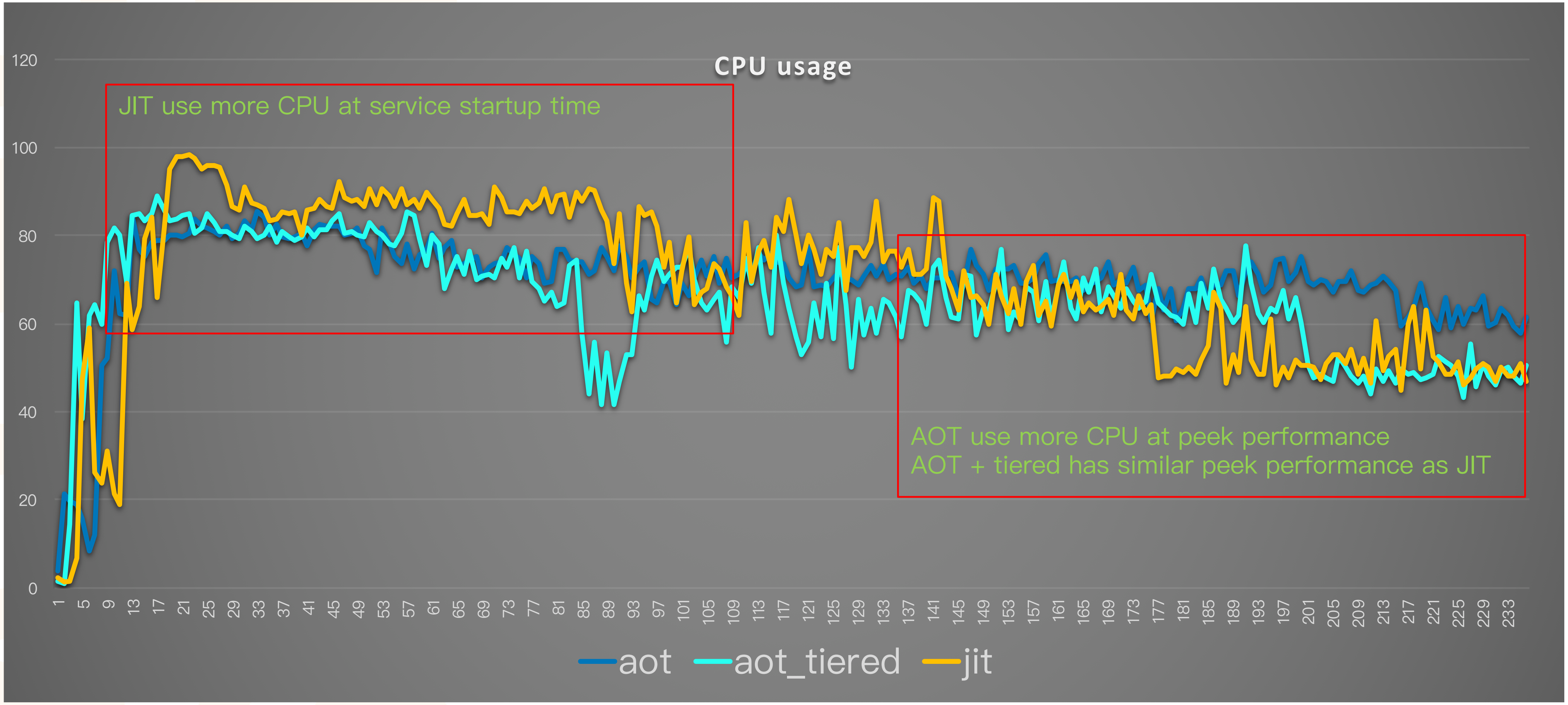


Normal JIT

Statistics

Requests	Executions			Response Times (ms)						Network (KB/sec)		
Label	#Samples	KO	Error %	Average	Min	Max	90th pct	95th pct	99th pct	Throughput	Received	Sent
Total	**2000000**	**0**	**0.00%**	**33.68**	**0**	**1095**	**38.00**	**46.00**	**55.00**	**2889.20**	**1503.26**	**648.94**
tpp_recommend_item	2000000	0	0.00%	33.68	0	1095	38.00	46.00	55.00	2889.20	1503.26	648.94

AOT performance — cpu usage



Future work

- Use profiling data to guide AOT compilation
- Integrate Dynamic AOT with AppCDS
- Try native-image.

Thank you!

Alibaba System Software