azul

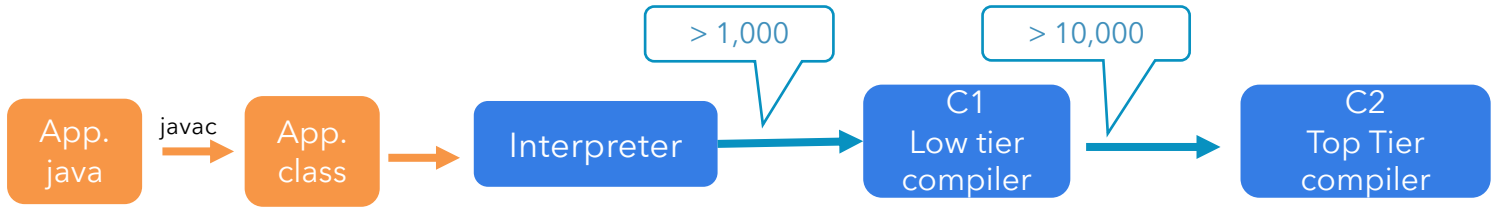
High Performance JVM

Satheesh Rajaraman

Vanilla Java Runtime(JVM) Limitations

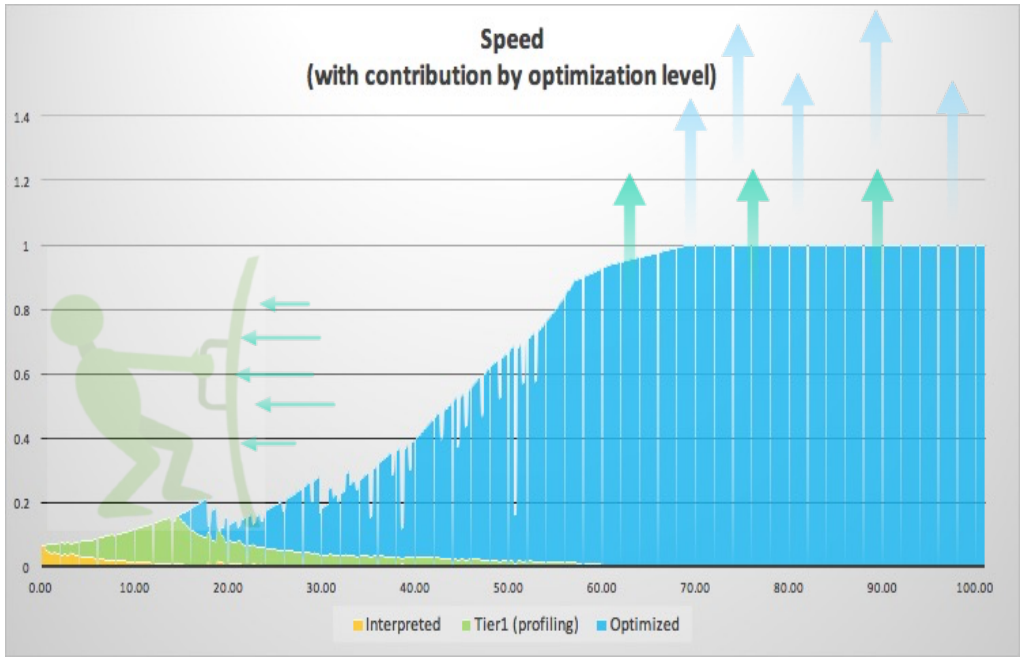
Lower Throughput

- Adaptive runtime JIT compilation:



Slow to Warmup

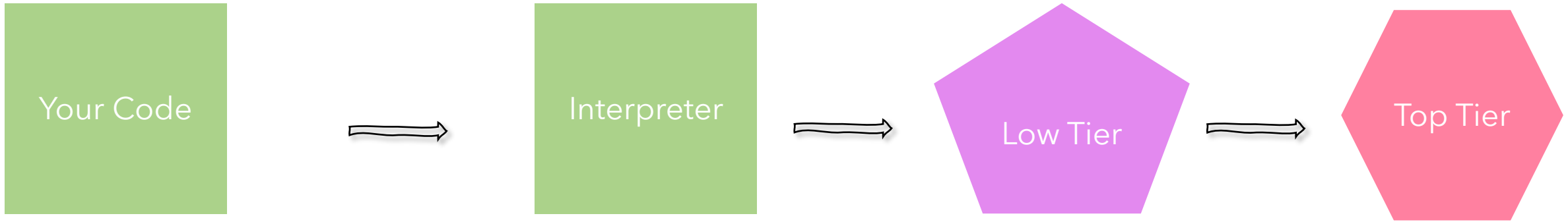
- Latency of the initial transactions after restart



azul

Demo

Source code vs JIT Compiled code



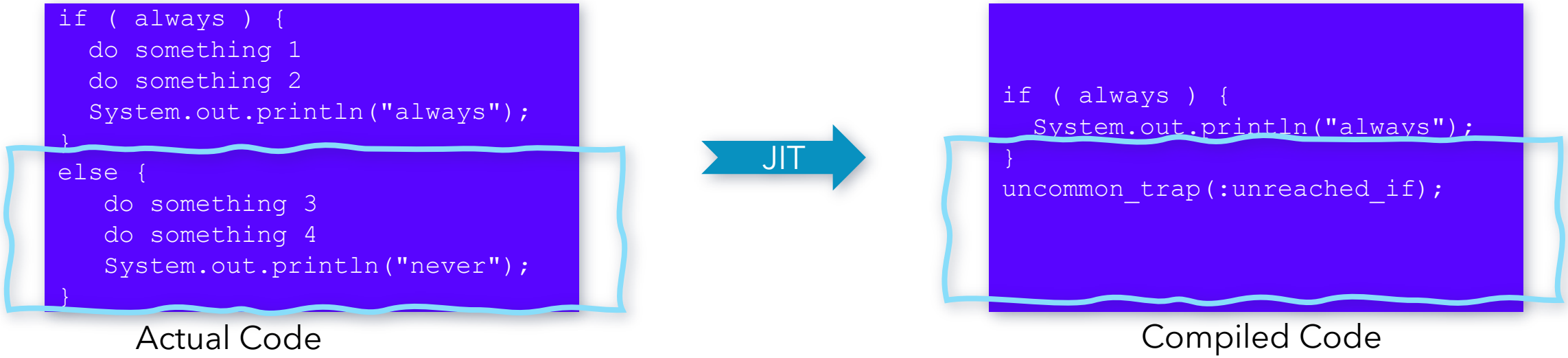
Your Code

Not Modified/optimized

Moderately modified/optimized

Heavily Modified/optimized

What is Speculative Optimization



What is Speculative Optimization

- Inlining
- Lock coarsening
- Loop unrolling
- Escape analysis
- Scalar replacement
- Branch Elimination
- Null Check Elimination
- Etc

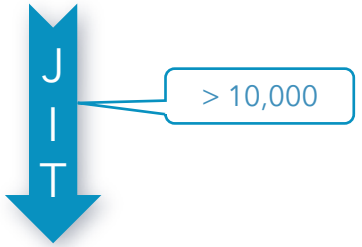
What is Speculative Optimization

- Inlining
- Lock coarsening
- Loop unrolling
- Escape analysis
- Scalar replacement
- Branch Elimination
- Null Check Elimination
- Etc..

```
public long multiply(long x, long y) {
    return x * y;
}

public void calculator() {
    long result = multiply(10, 20);
}
```

Actual Code




```
public void calculator() {
    long result = 10 * 20;
}
```

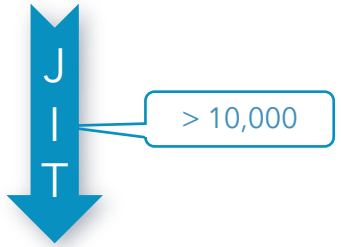
Compiled Code

What is Speculative Optimization

- Inlining
- Lock coarsening
- Loop unrolling
- Escape analysis
- Scalar replacement
- Branch Elimination
- Null Check Elimination
- Etc..

```
for(int i = 0; i < numIterations; i++) {
    METRICS.gauge("test.metric", i);
}
```

Actual Code




```
for(int i = 0; i < numIterations; i += 5) {
    METRICS.gauge("test.metric", i);
    METRICS.gauge("test.metric", i+1);
    METRICS.gauge("test.metric", i+2);
    METRICS.gauge("test.metric", i+3);
    METRICS.gauge("test.metric", i+4);
}
```

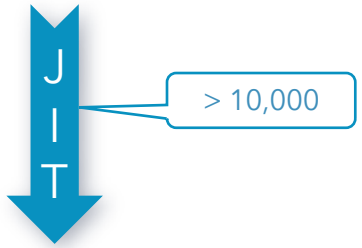
Compiled Code

What is Speculative Optimization

- Inlining
- Lock coarsening
- Loop unrolling
- Escape analysis
- Scalar replacement
- Branch Elimination
- Null Check Elimination
- Etc..

```
private static void runSomeAlgorithm(Graph graph) {
            if (graph == null) {
                    return;
             }
            // do something with graph
}
```

Actual Code




```
private static void runSomeAlgorithm(Graph graph) {
             // do something with graph
}
```

Compiled Code

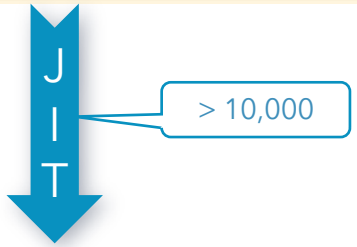
What is Speculative Optimization

- Inlining
- Lock coarsening
- Loop unrolling
- Escape analysis
- Scalar replacement
- Branch Elimination
- Null Check Elimination
- Etc..

```
public static void doBranch(int branch){
  if (branch == 1)  y=1;
  else if (branch == 2) y=2;
  else if (branch == 3) y=3;
}

for(int i=0; i>20000; i++){
  doBranch(1);
}
```

Actual Code

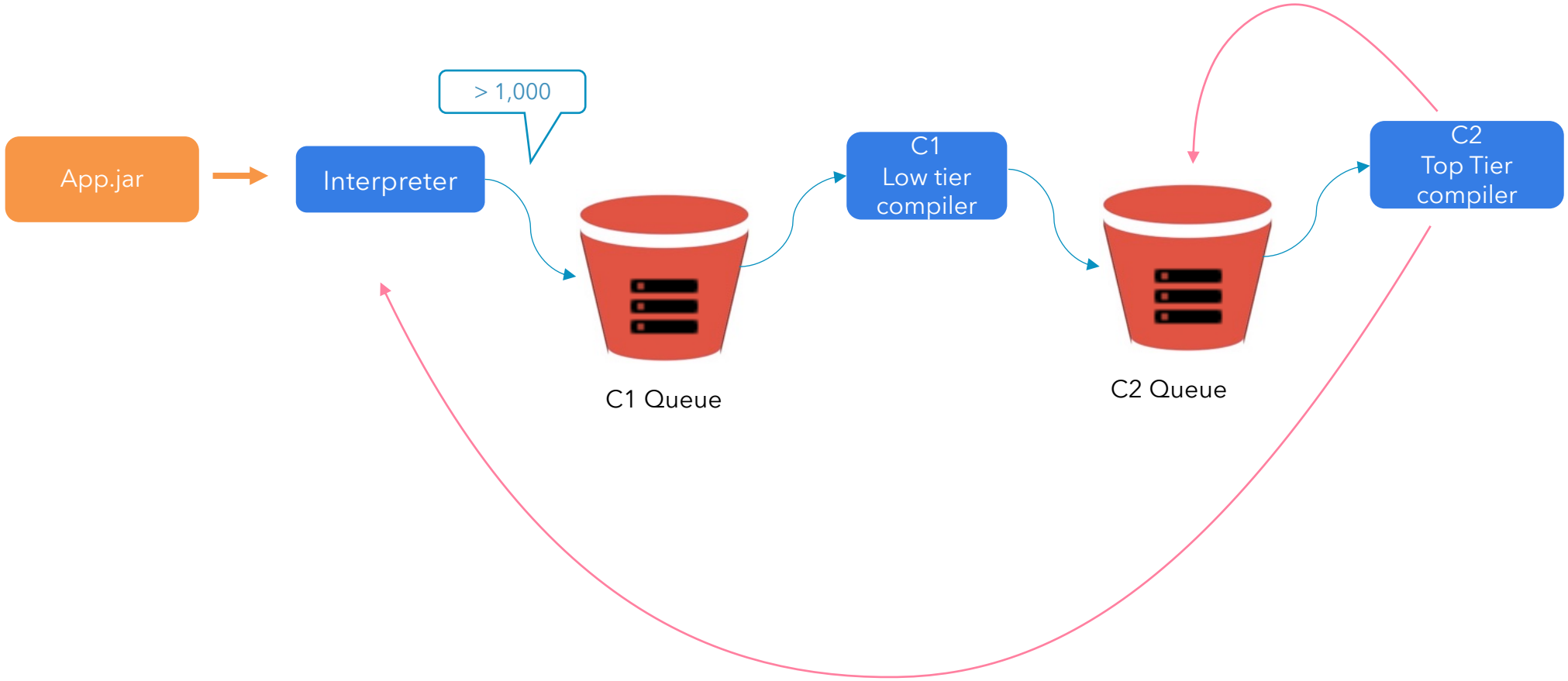



```
public static void doBranch(int branch){
  if (branch == 1)  y=1;
  uncommon_trap(:unreached_if);
}

for(int i=0; i>20000; i++){
  doBranch(1);
}
```

Compiled Code

What happens when assumption goes wrong ?



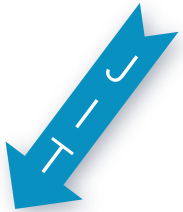
What happens when assumptions goes wrong ?

```
public static void doBranch(int branch){
  if (branch == 1)  y=1;
  else if (branch == 2) y=2;
  else if (branch == 3) y=3;
}

for(int i=0; i>20000; i++){
  doBranch(1);
}
```

Actual Code

Before

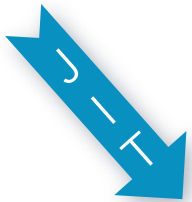


Compiled Code

```
public static void doBranch(int branch){
  if (branch == 1)  y=1;
  uncommon_trap(:unreached_if);
}

for(int i=0; i>20000; i++){
  doBranch(1);
}
```

After



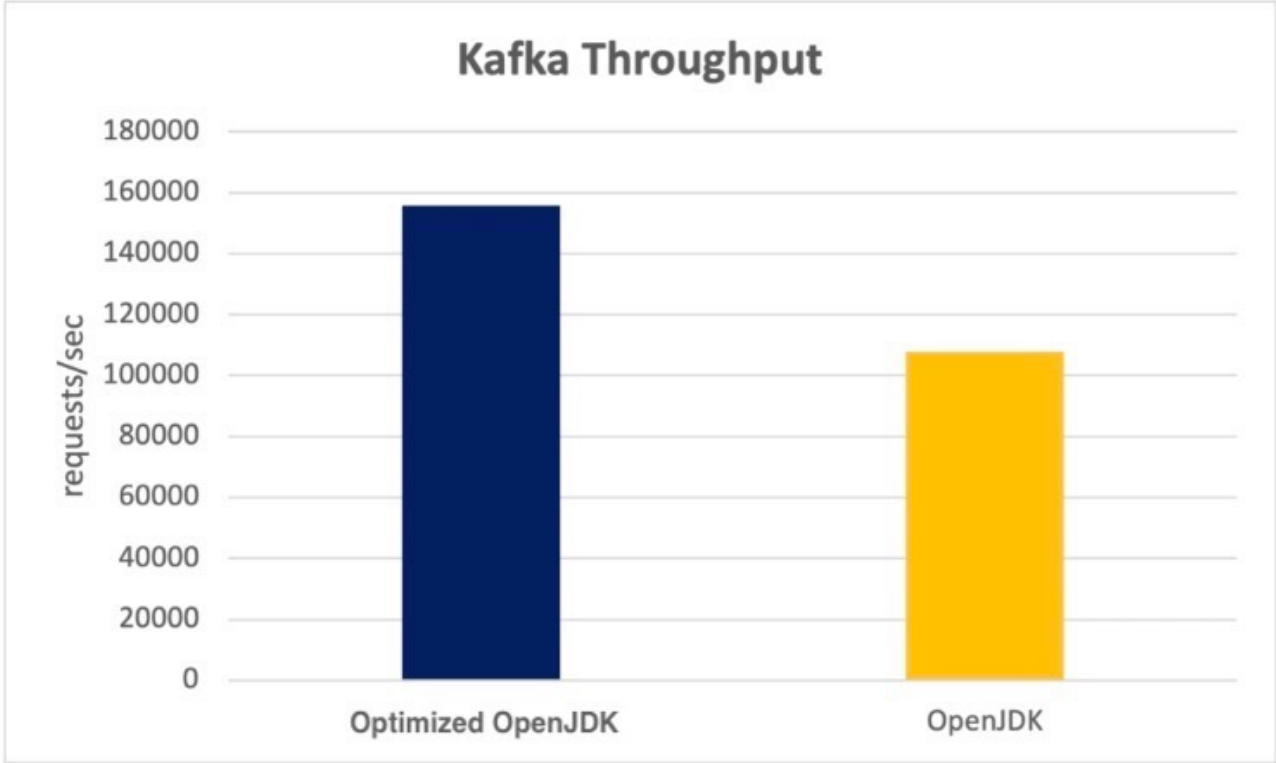
Re-Compiled Code

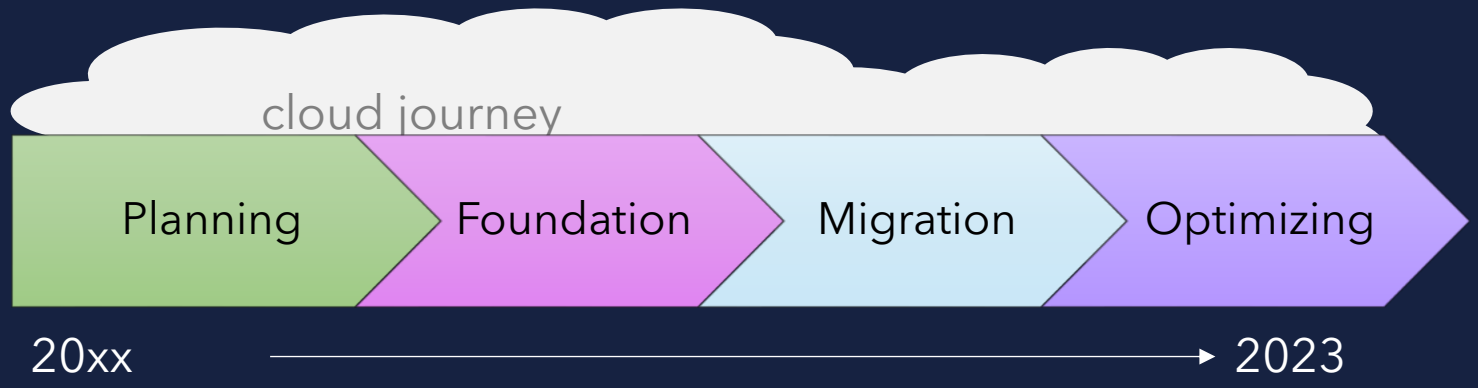
```
public static void doBranch(int branch){
  if (branch == 1)  y=1;
  else if (branch == 2) y=2;
  uncommon_trap(:unreached_if);
}

for(int i=0; i>20000; i++){
  doBranch(1);
}
```

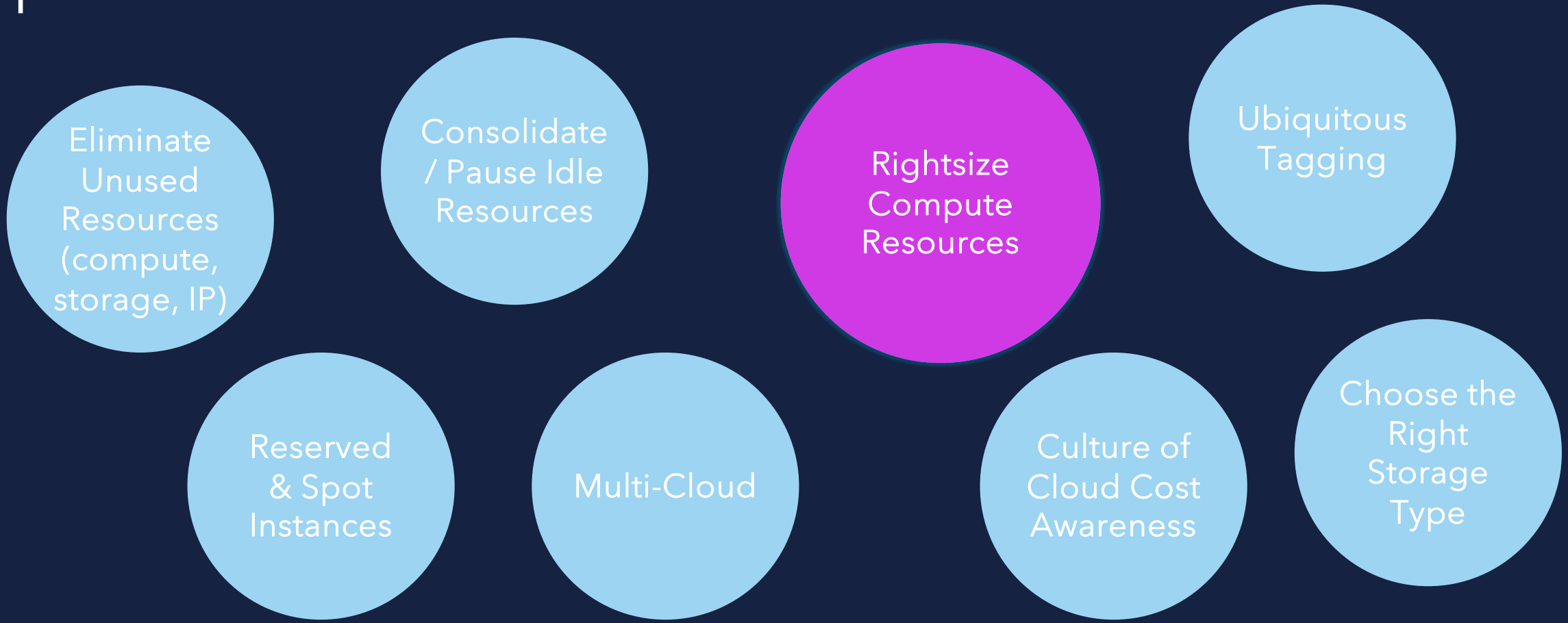
Kafka Max Throughput (requests/sec)

Optimized OpenJDK	155,797
Vanilla OpenJDK	107,805





Widely Adopted Approaches to Optimize & Monitor Cloud Costs



"Perhaps the largest opportunity in infrastructure right now is sitting somewhere between cloud hardware

...and the unoptimized code running on it."

The Cost of Cloud, a Trillion Dollar Paradox
by Sarah Wang and Martin Casado (May 2021)

andreessen. horowitz

High Performance

Enhanced Build of OpenJDK

Faster Apps = Less Compute = Lower Cloud Bill

start fast
go fast
stay fast

Enterprises Reducing Cloud Costs

Taboola

30%+ reduction in server footprint

translating to millions of dollars in savings

bazaarvoice

Would be running 5x infrastructure

without Azul Platform Prime

SUPERCELL

Reclaimed 20%+ in unused CPU

carrying capacity, improving gamer experience

arek

85% reduction in cost

per transaction with cloud-based financial system

zoominfo

Reduced infrastructure spend by 38%

through improved performance

BIDS TRADING

Reclaimed 50%

Of unused infrastructure used for electronic trading

Maximize Efficiency, Optimize Cost

azul
The Java Platform for the Modern Cloud Enterprise

kafka

Higher Volume Ingestion

45% more efficient

Infra reduction for cloud savings

Faster streaming for microservices

Qualys.

25% infra & 20% infra spend reduction

Cassandra

Infrastructure optimization

2-5x more efficient under load

More efficient storage growth

Cloud cost savings

improve digital by azerion

5X Improvement in TPS, 20ms SLA

Solr elastic

Transform Search Experience

1.5-5x more efficient under load

30% faster catalog, mktg search

zoominfo

147% TPS inc, 38% cloud spend reduction

Improve your Digital Transformation Projects; Allow you to grow with Confidence

Thank You.