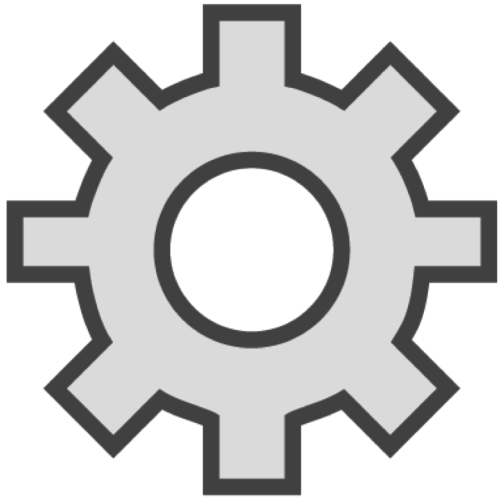
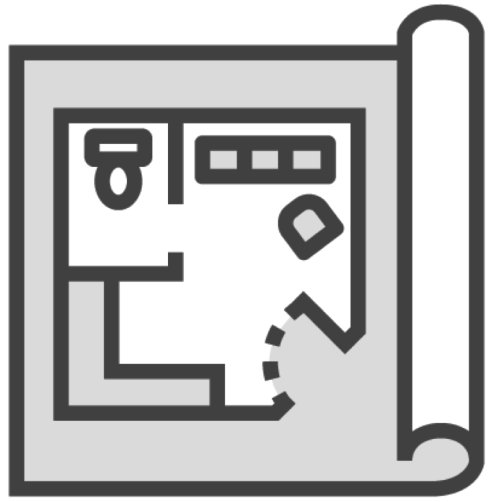
Performance Factors





Overview

- Understand your performance characteristics
- Java application profiling
- JVM tuning
- Search and data structures
- Memory management
- Optimizing concurrent code
- Avoid doing expensive operations

Performance and Computer Hardware

Computer Performance

Is the amount of work accomplished by a computer system

Requests per second

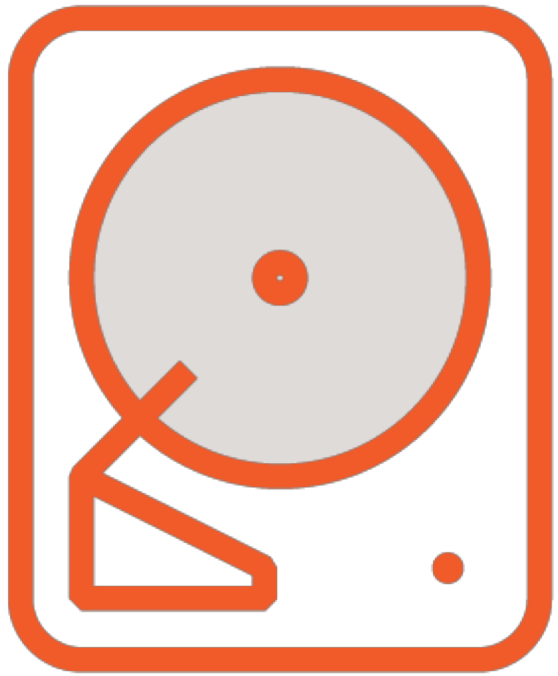
Queries per second

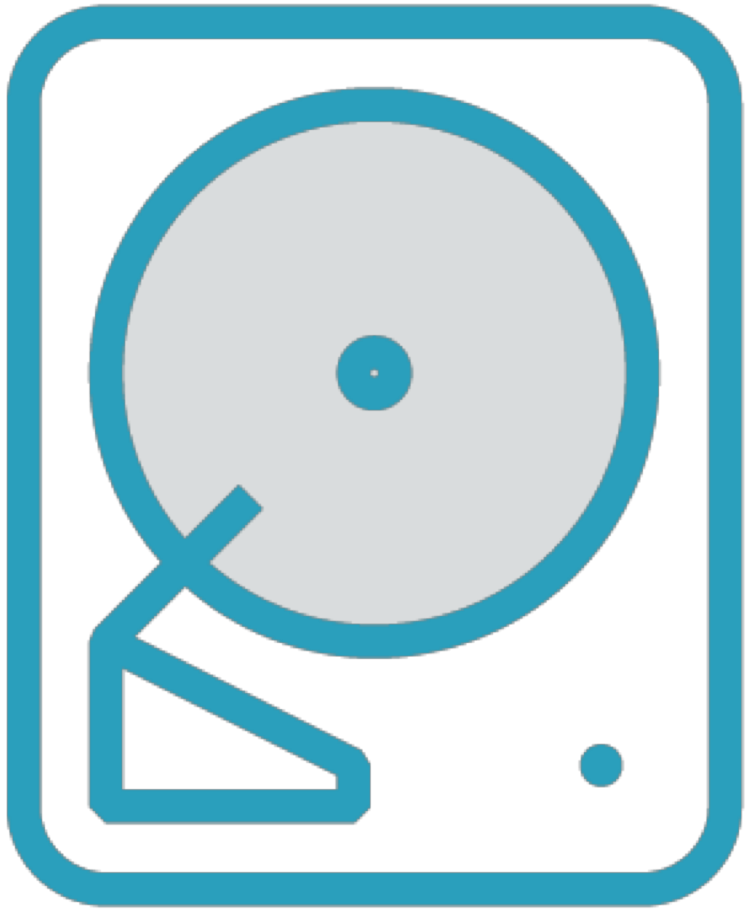
Frames per second

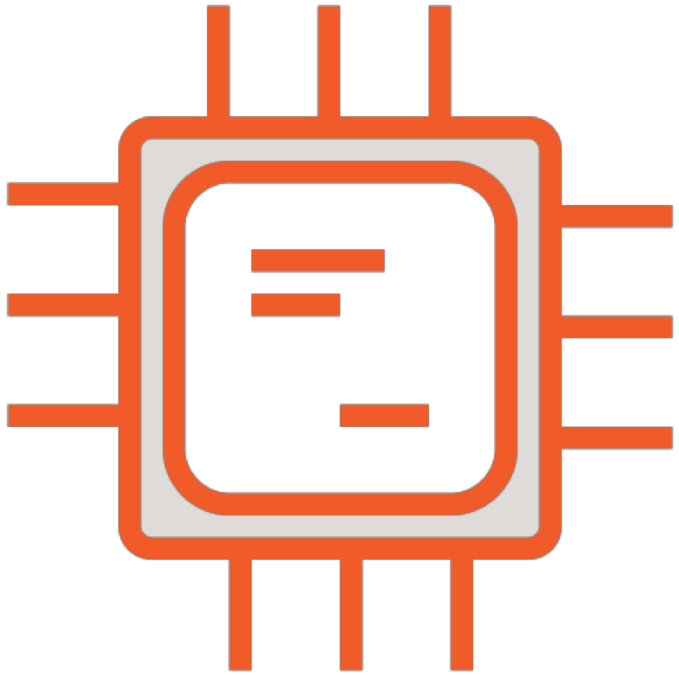
Operations per second

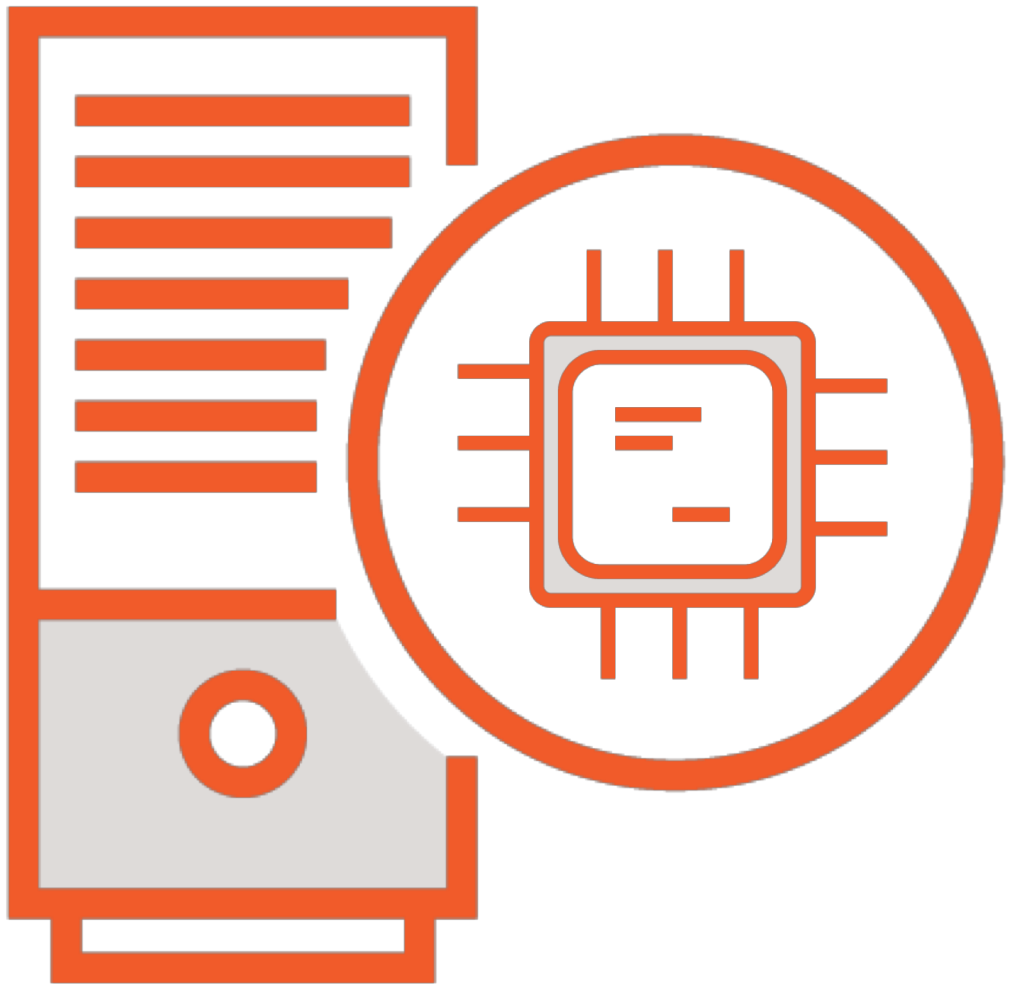
Instructions per second

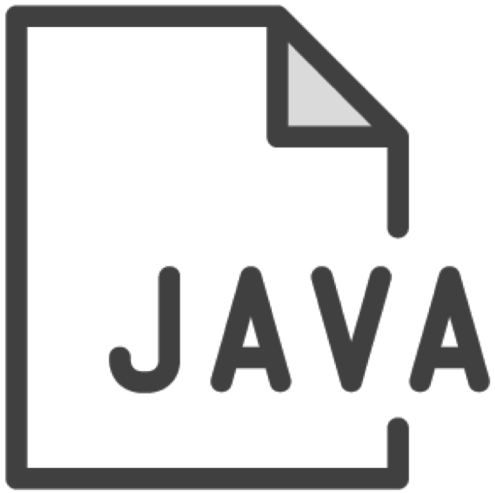
The better the machine, the more instructions it can execute per second





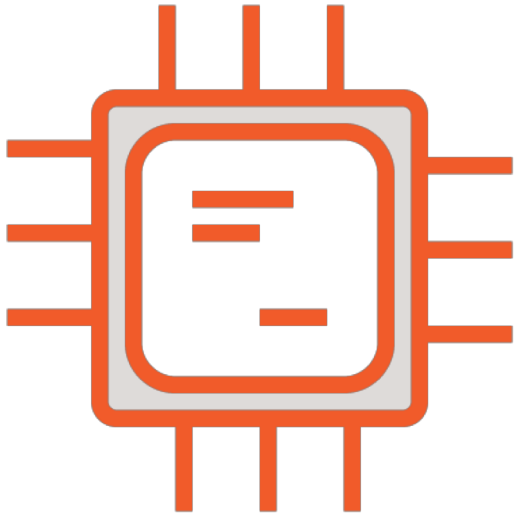




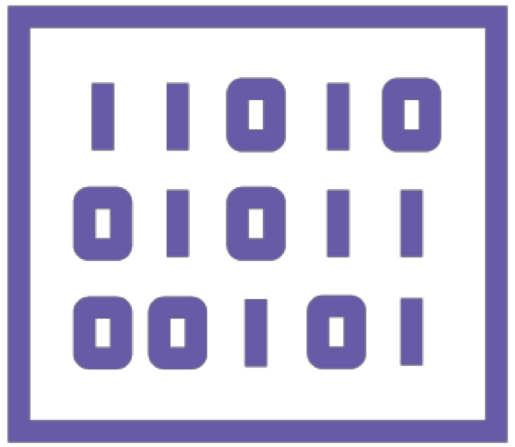


```
MOV AX, CX
SUB CX, BX
CMP BX, 300H
JLE NEXT
INC AX
...
```

Hardware Components

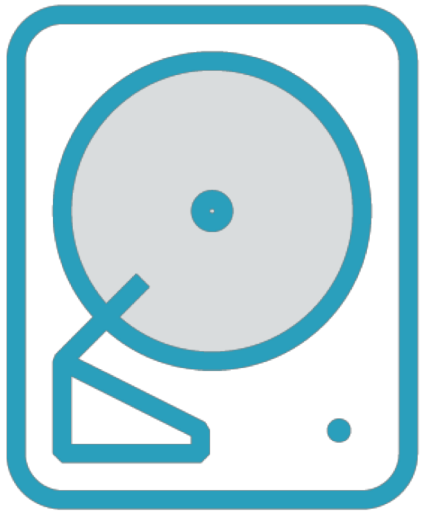


CPU

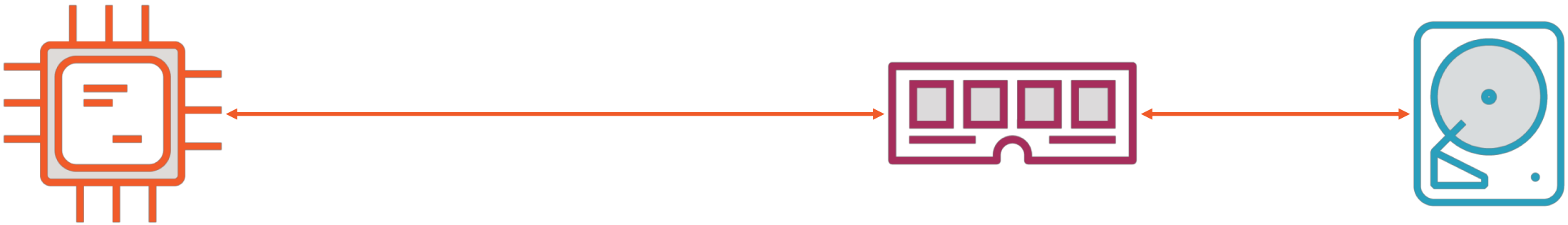


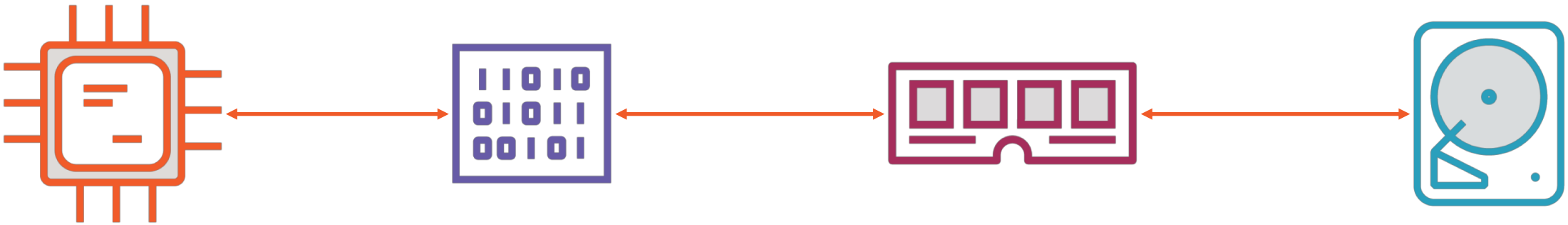
Cache

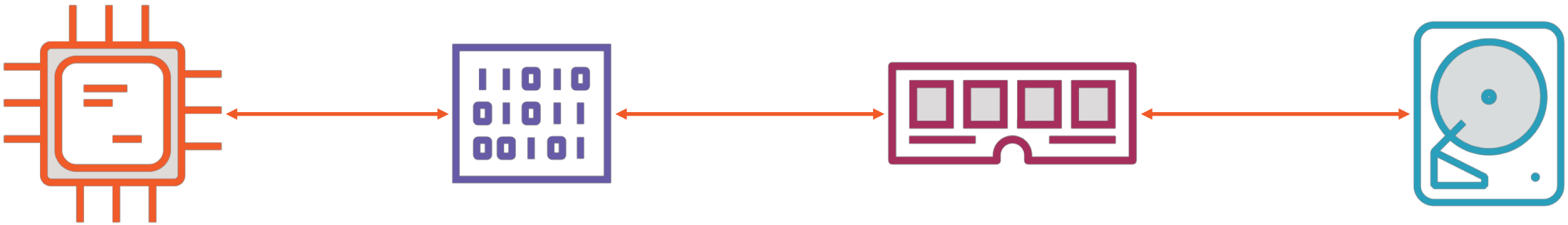
RAM

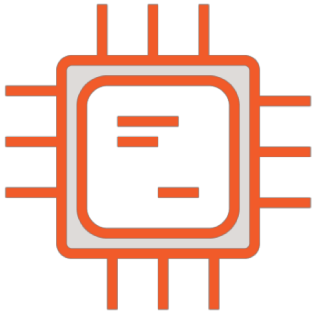
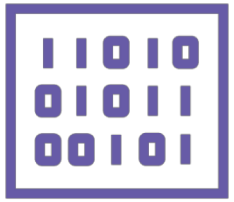


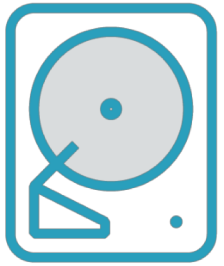
I/O

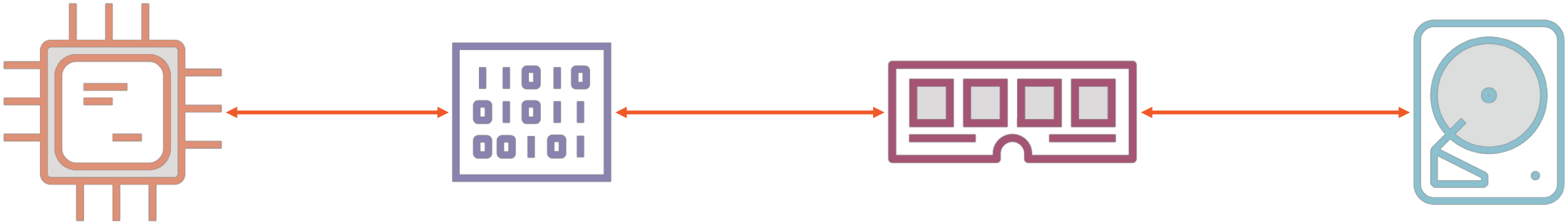




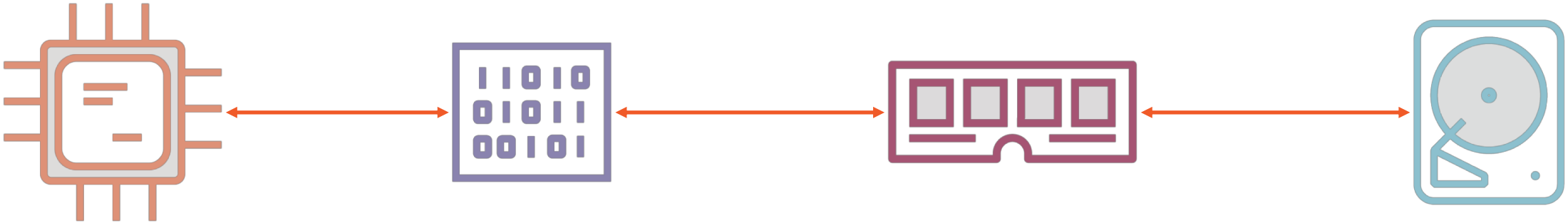






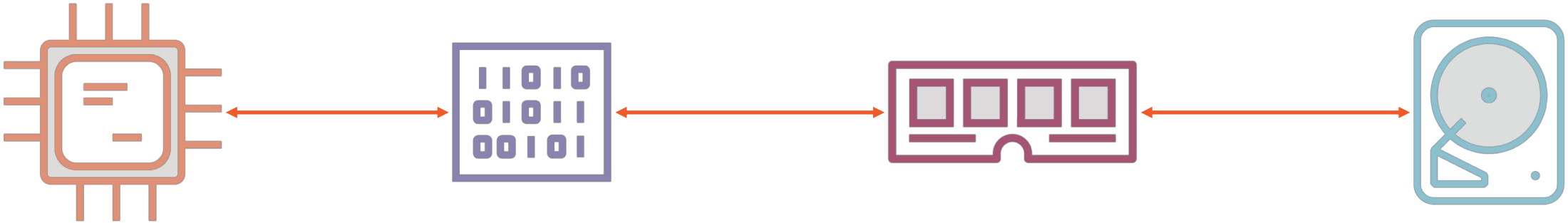
CPU



CPU

L1i/L1d

32/32 KB

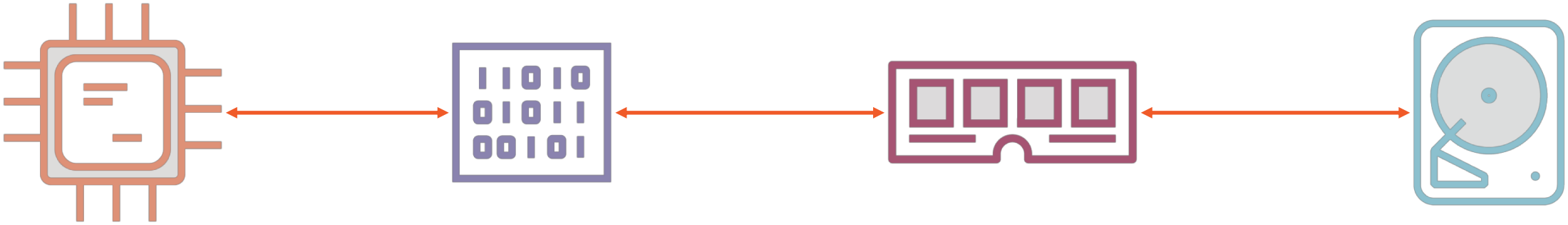


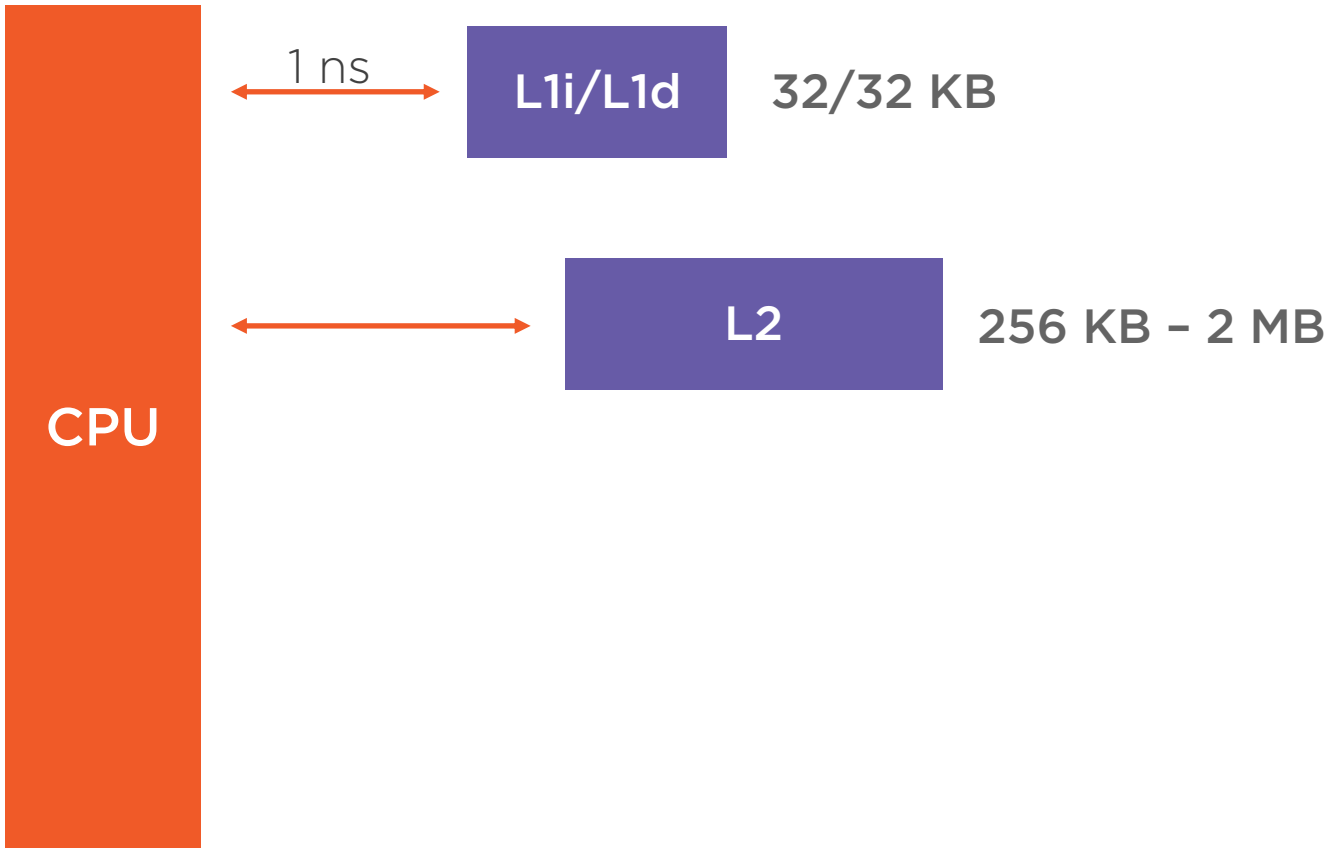
CPU

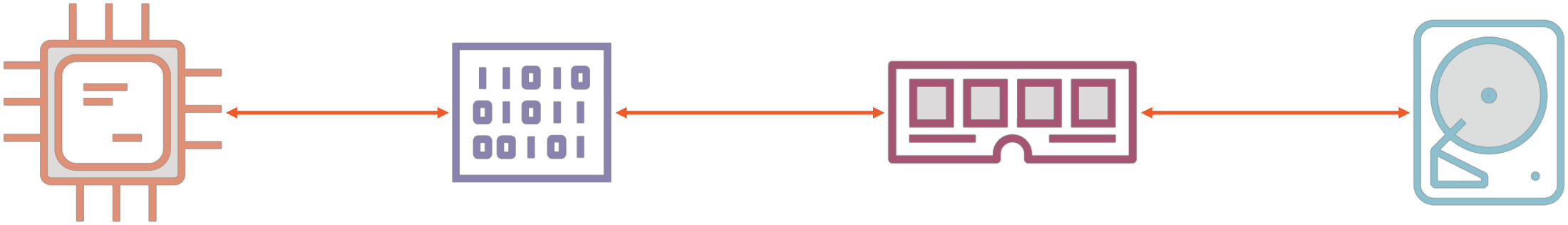
1 ns

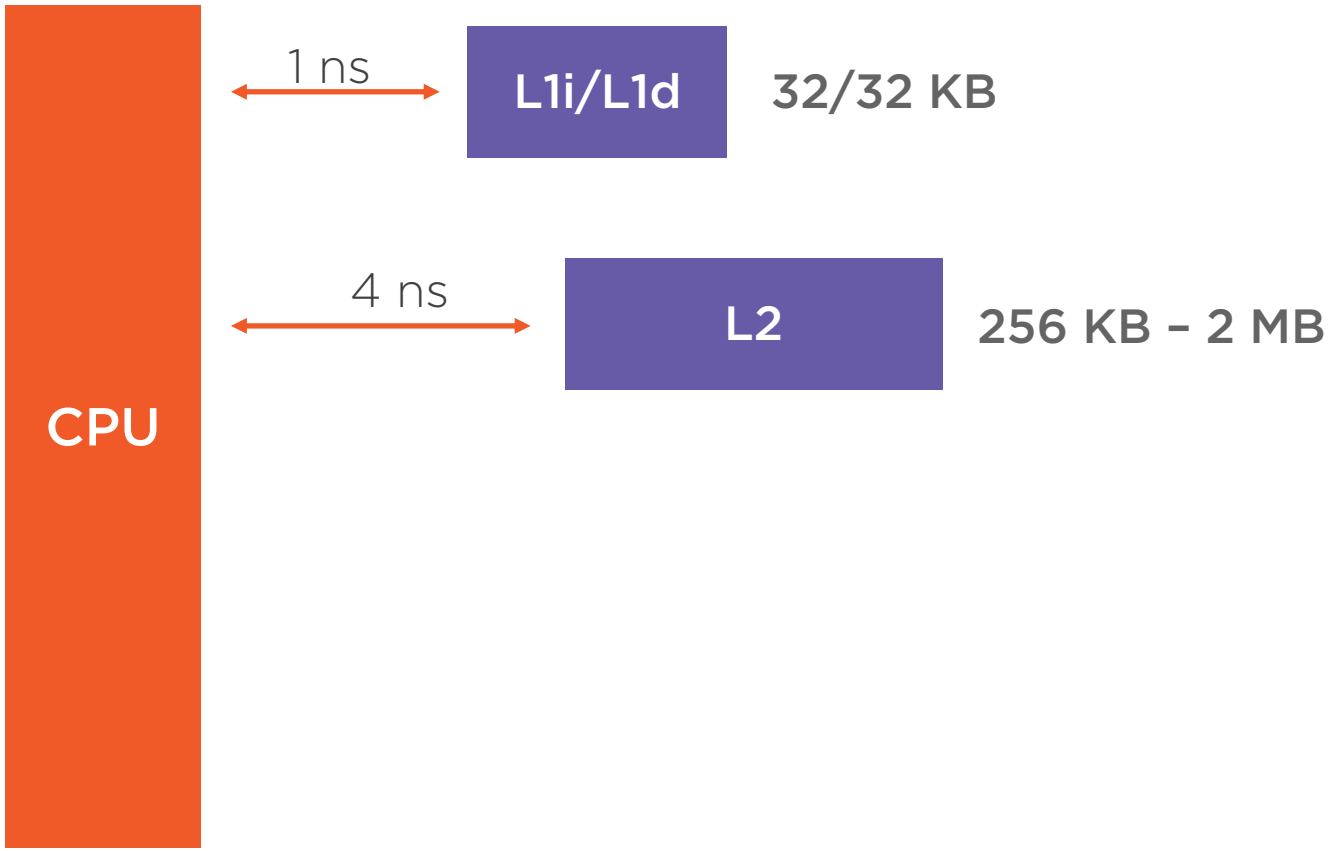
L1i/L1d

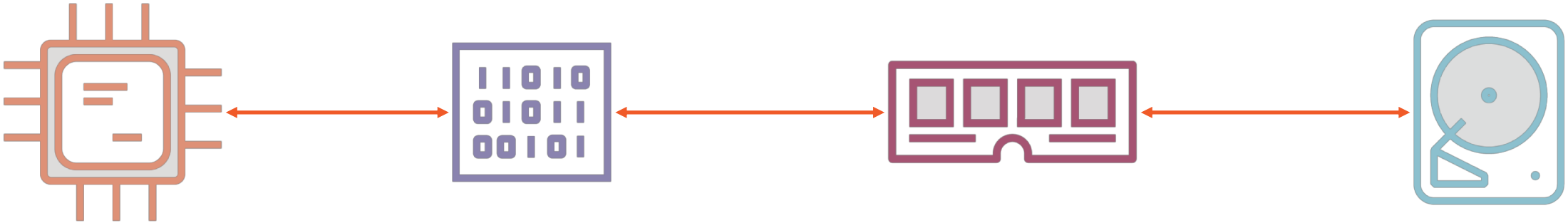
32/32 KB

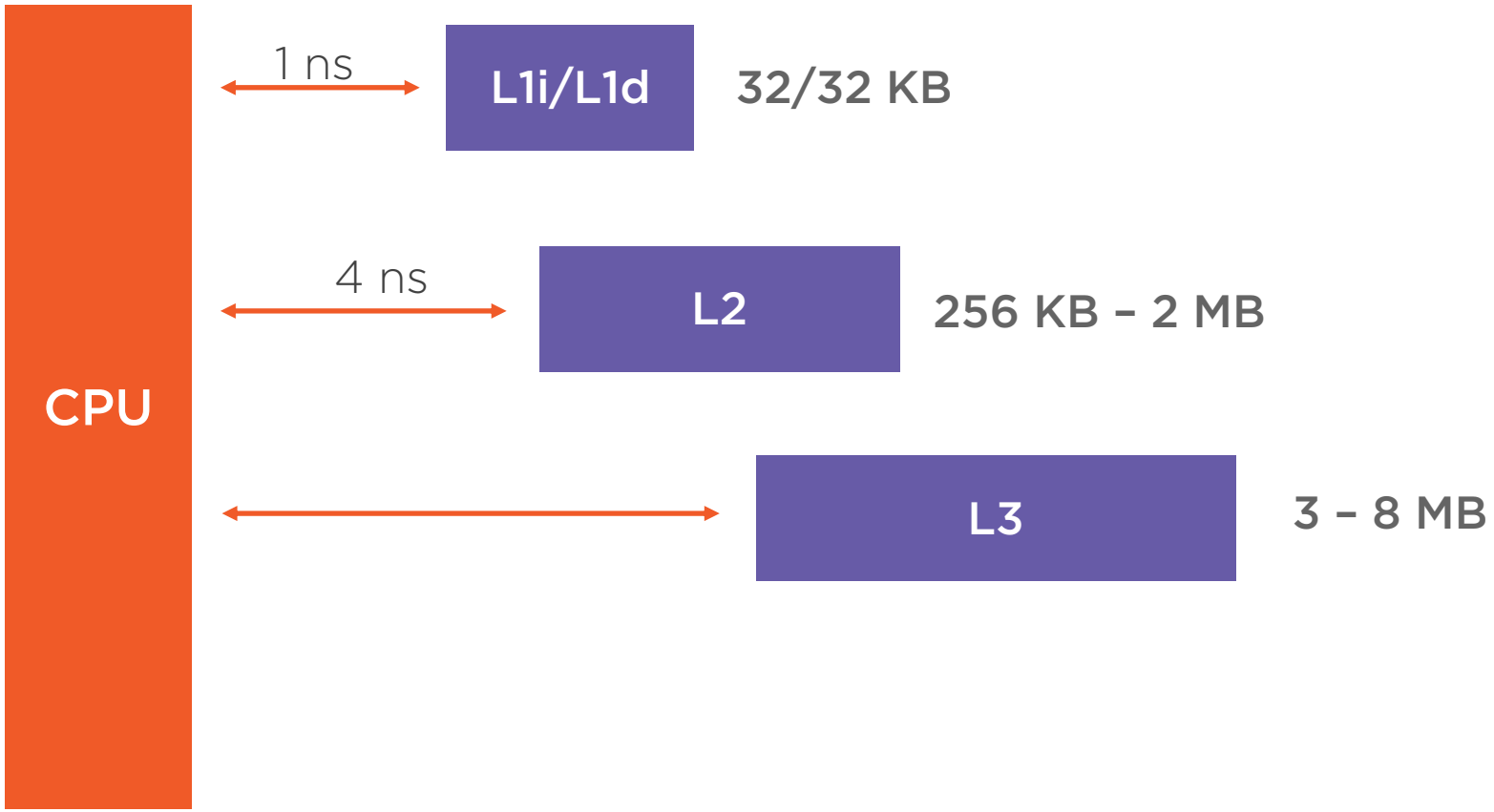


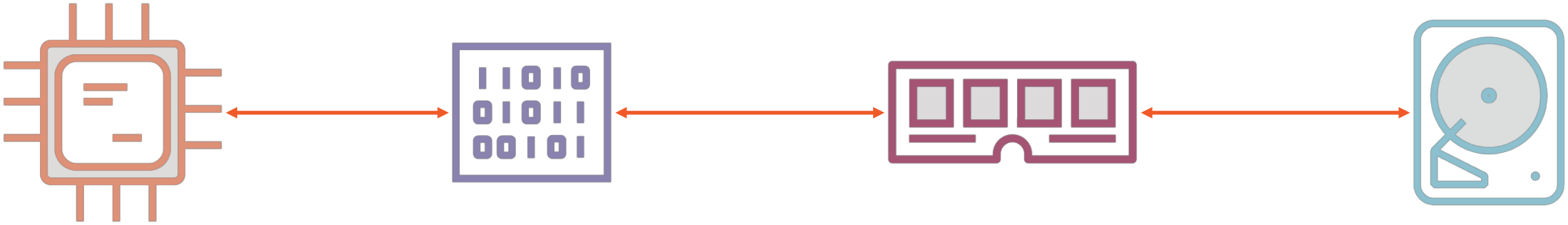

CPU
1 ns
L1i/L1d
32/32 KB
L2
256 KB – 2 MB

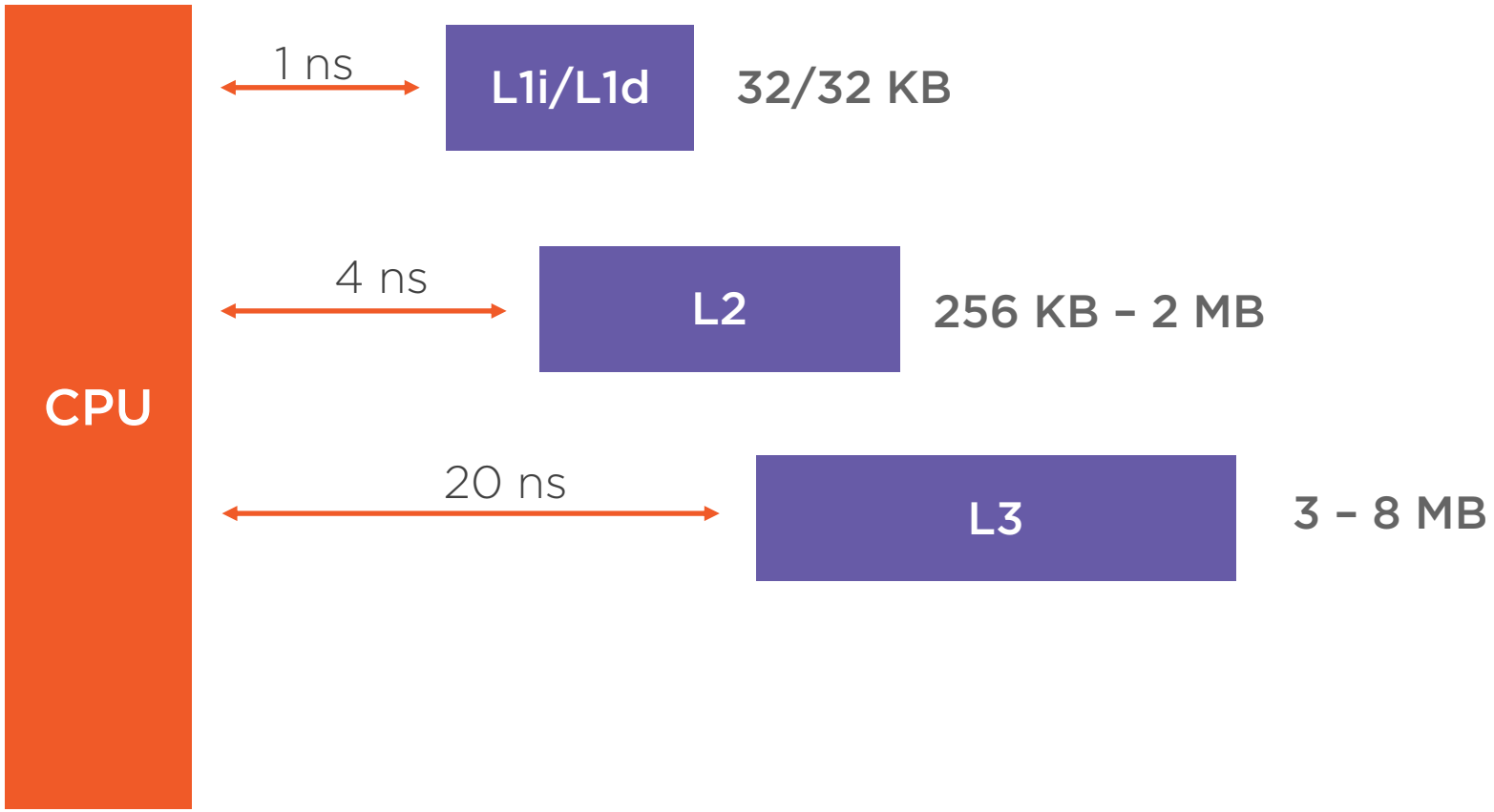


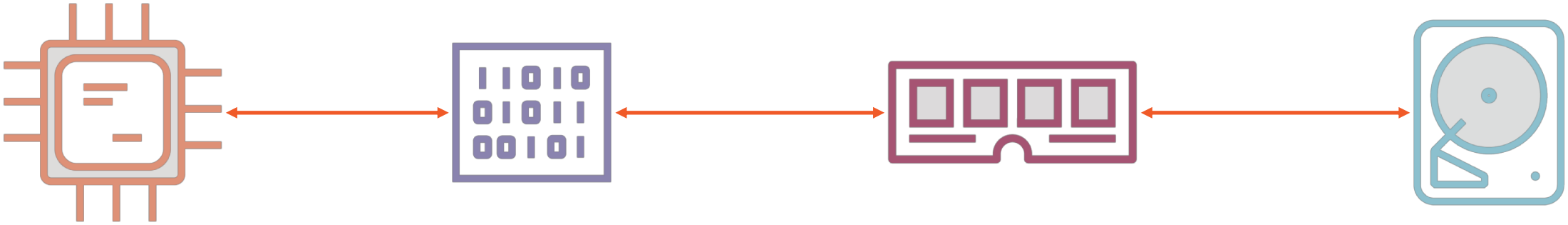

CPU
1 ns
L1i/L1d
32/32 KB
4 ns
L2
256 KB – 2 MB

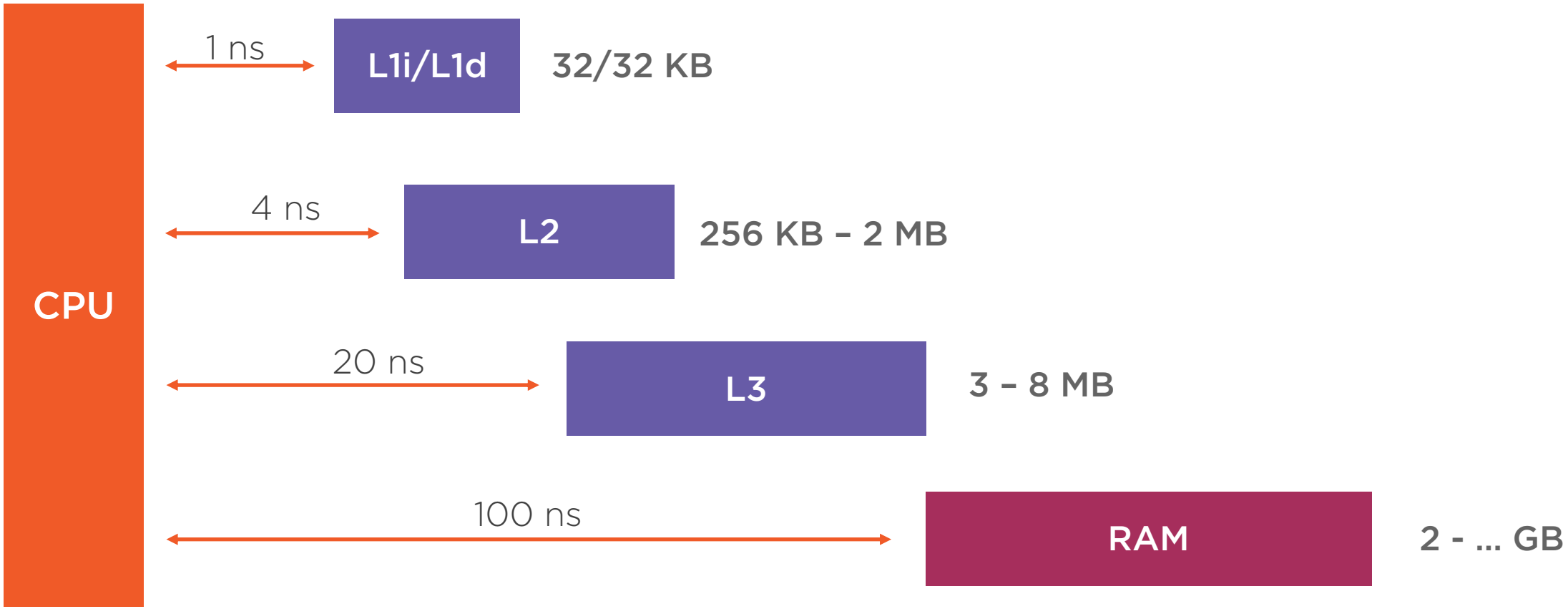



CPU
1 ns
L1i/L1d
32/32 KB
4 ns
L2
256 KB – 2 MB
L3
3 – 8 MB




CPU
1 ns
L1i/L1d
32/32 KB
4 ns
L2
256 KB – 2 MB
20 ns
L3
3 – 8 MB




CPU
1 ns
L1i/L1d
32/32 KB
4 ns
L2
256 KB – 2 MB
20 ns
L3
3 – 8 MB
100 ns
RAM
2 - ... GB

The smaller the program is, the higher the likelihood of it's instructions and data fitting in cache

https://people.eecs.berkeley.edu/~rcs/research/interactive_latency.html

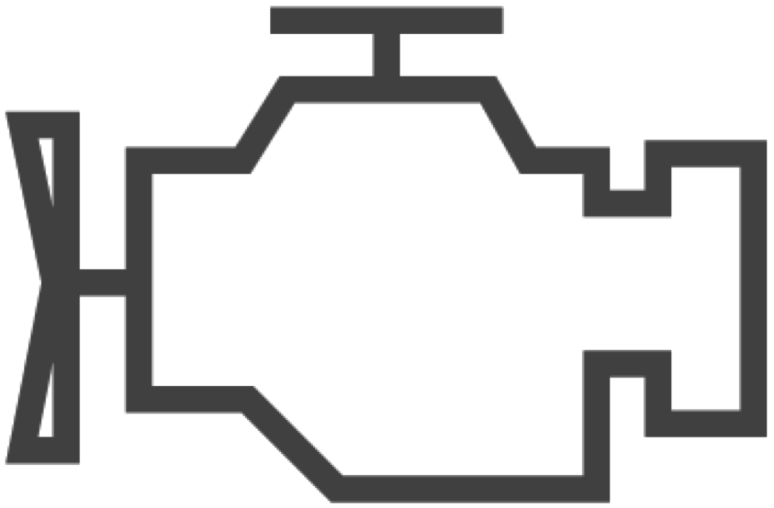
Main memory	Reference	100	Reading 1 million bytes	5000
SSD	Random Access	16,000	Reading 1 million bytes	78,000
HDD	Disk Seek	3,000,000	Reading 1 million bytes	1,000,000

Sequential data access is generally faster than random access

System Performance Metrics

Utilization, Saturation and Errors method (USE)

Brendan Gregg



CPU cycle

RAM capacity

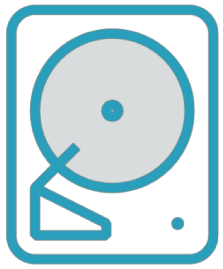
Disk capacity

Disk I/O

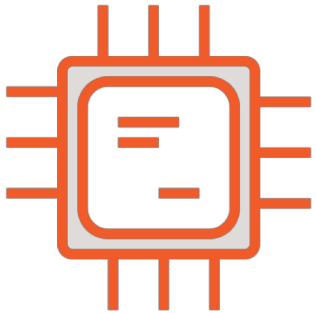
Network I/O

Utilization

Is the proportion of a resource that is used or the average time that the resource was busy servicing work



20%



50%

Saturation

Is the degree to which the resource has extra work that it can't service

CPU run queue

Swap space

Errors

Is the count of error events for a resource

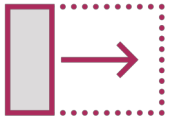
Effect of Errors on Performance

Error handling

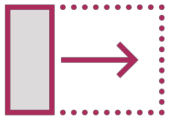
Retries

Fewer pool resources

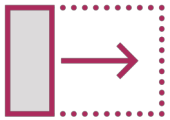
Software Resources



Thread pools



Locks



Threads / File descriptors

USE Methodology

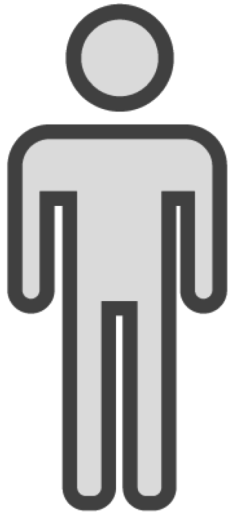
	Utilization	Saturation	Errors
CPU	☐	☐	☐
Memory	☐	☐	☐
Disk Capacity	☐	☐	☐
Disk I/O	☐	☐	☐
Network I/O	☐	☐	☐

USE Methodology

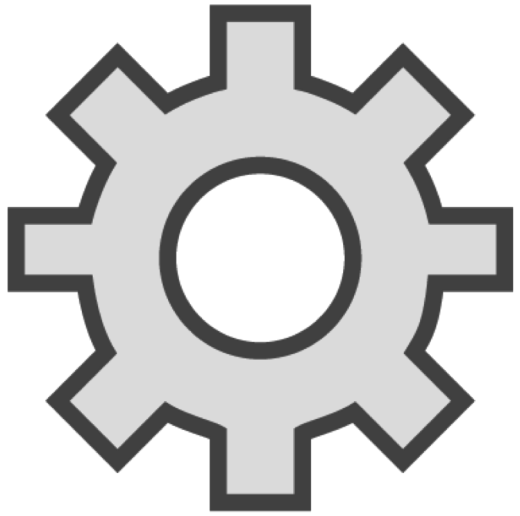
	Utilization	Saturation	Errors
CPU			☐
Memory	☐		☐
Disk Capacity		☐	☐
Disk I/O			
Network I/O		☐	

Performance Testing Tools - OS

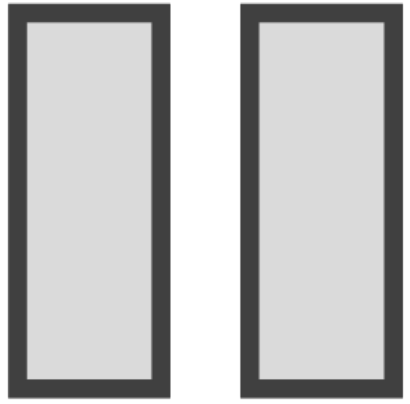
CPU Utilization



User Time



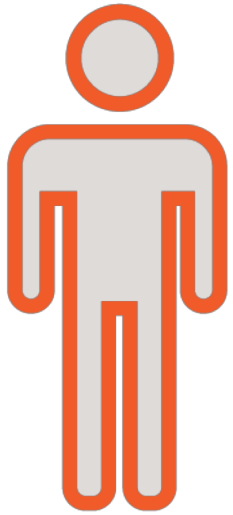
System Time



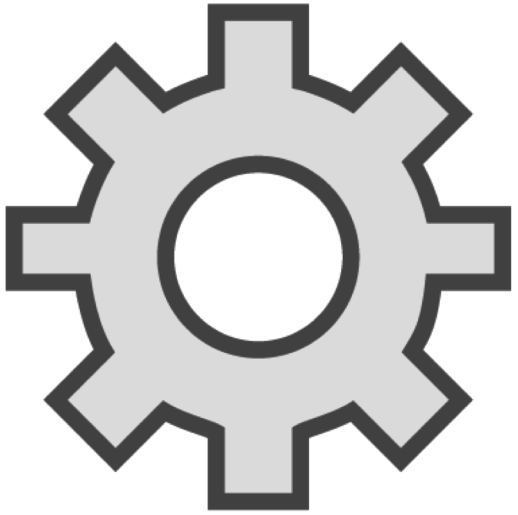
Idle Time

I/O Wait Time

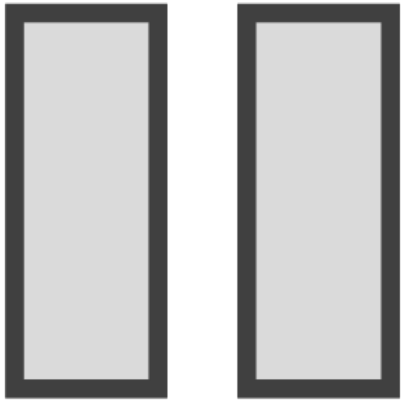
CPU Utilization



User Time



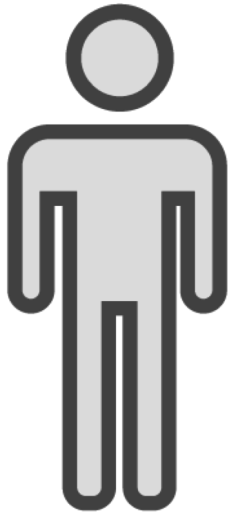
System Time



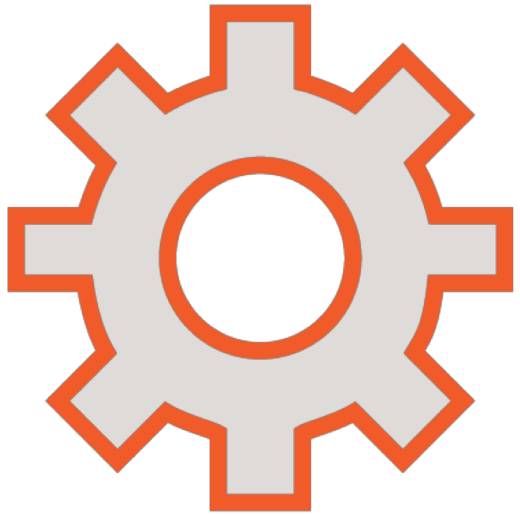
Idle Time

I/O Wait Time

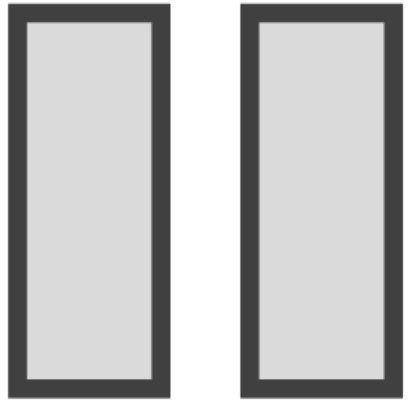
CPU Utilization



User Time

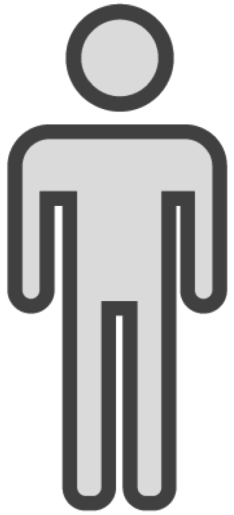


System Time

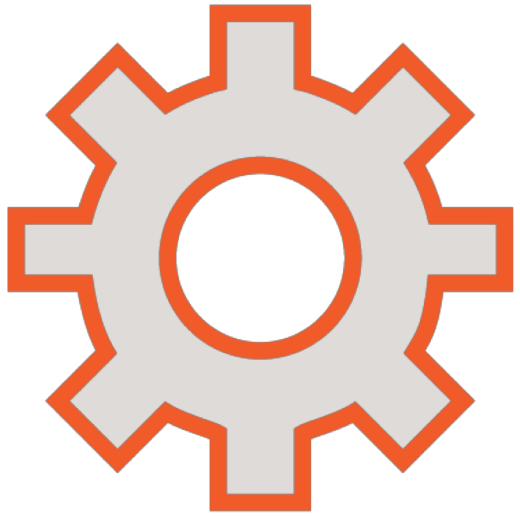


Idle Time

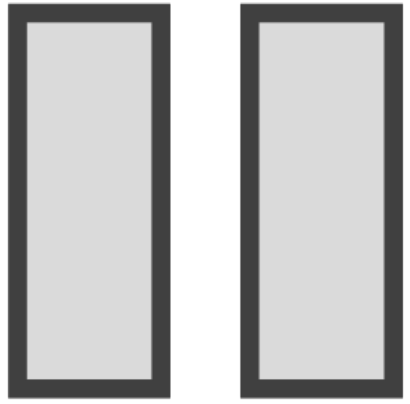
CPU Utilization



User Time



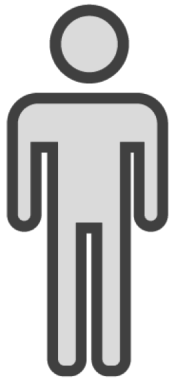
System Time



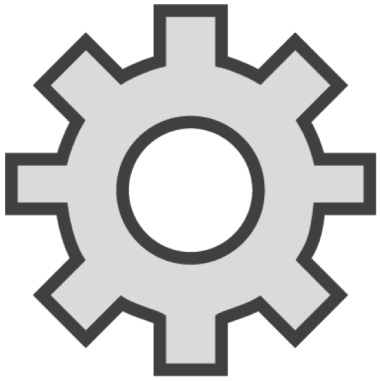
Idle Time

I/O Wait Time

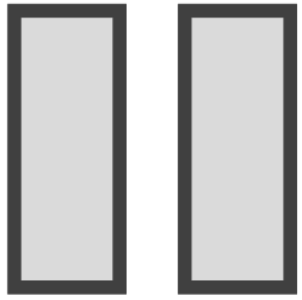
CPU Utilization: 2%



2%



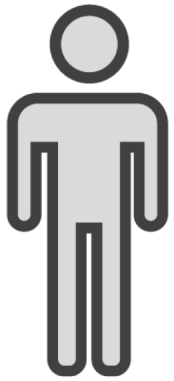
0%



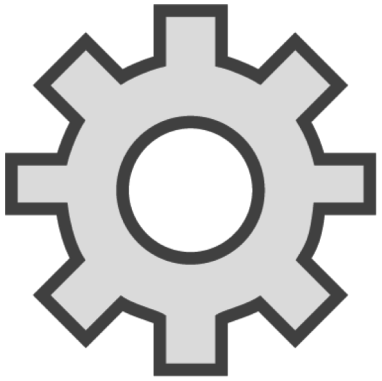
98%

0%

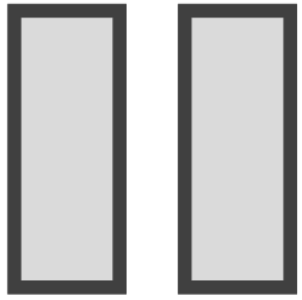
CPU Utilization: 2%



20 ms



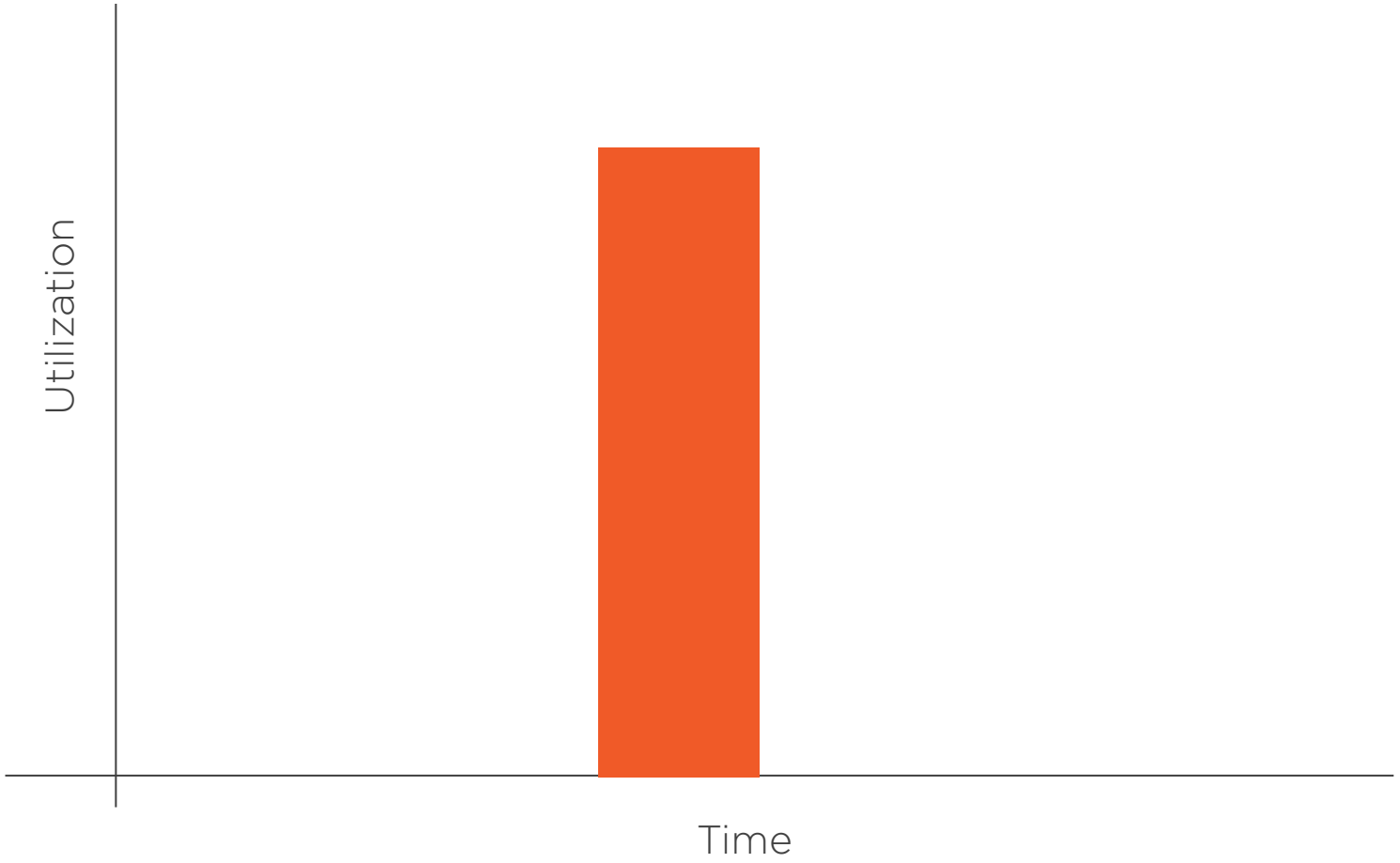
0 ms



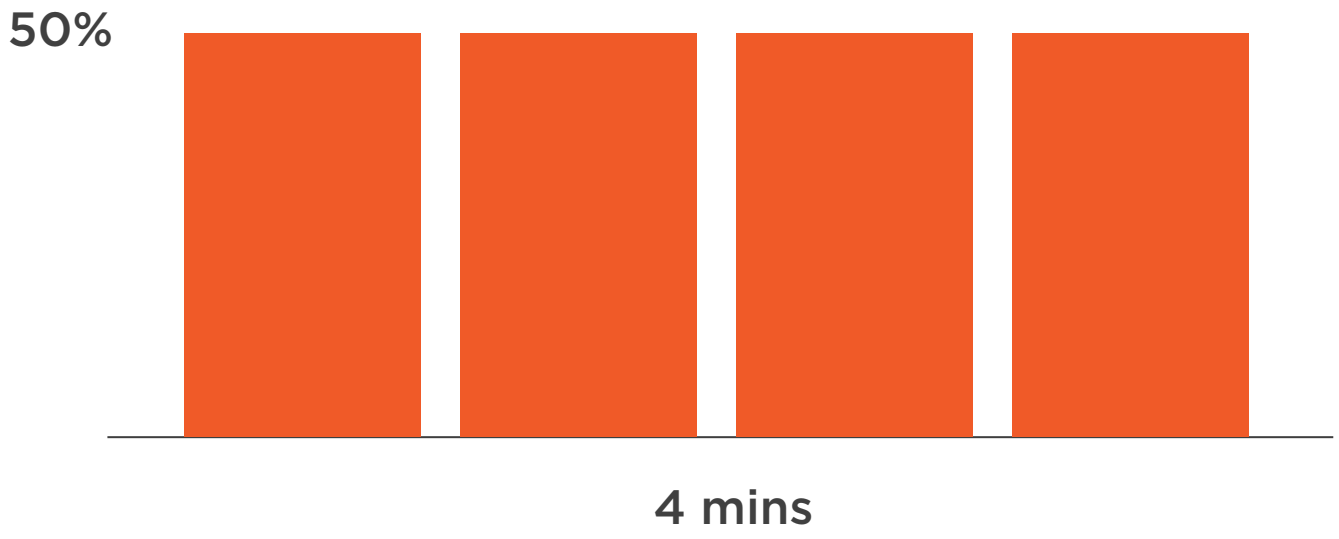
980 ms

0 ms

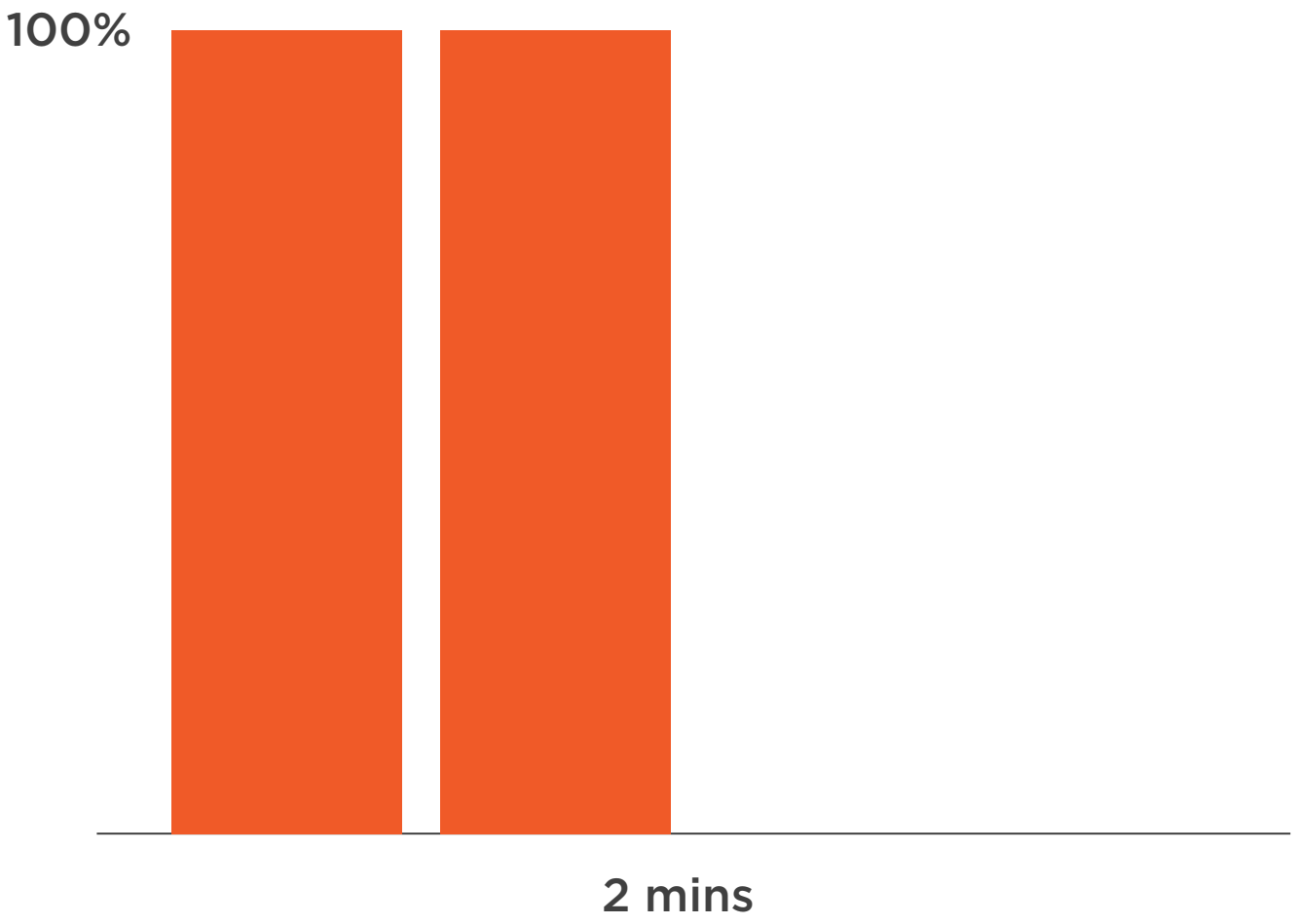
CPU Utilization



CPU Utilization



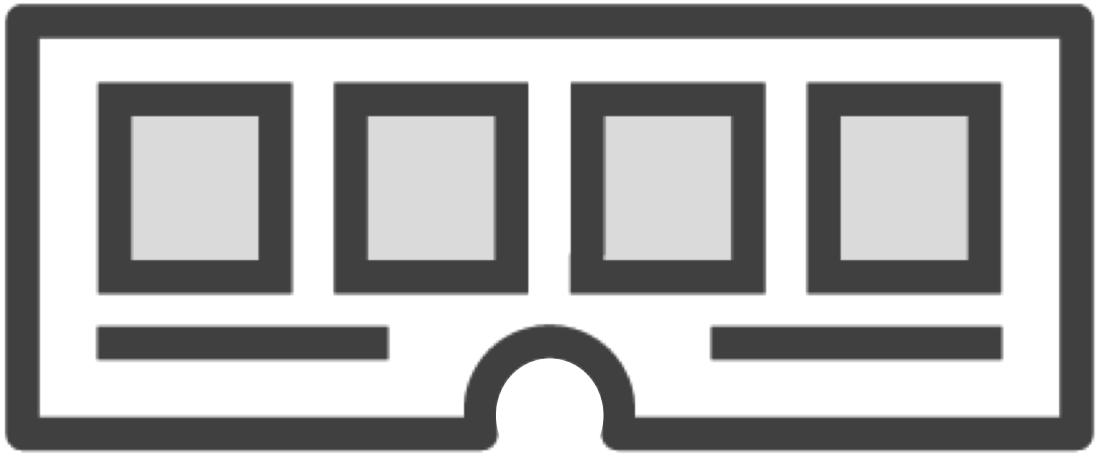
CPU Utilization

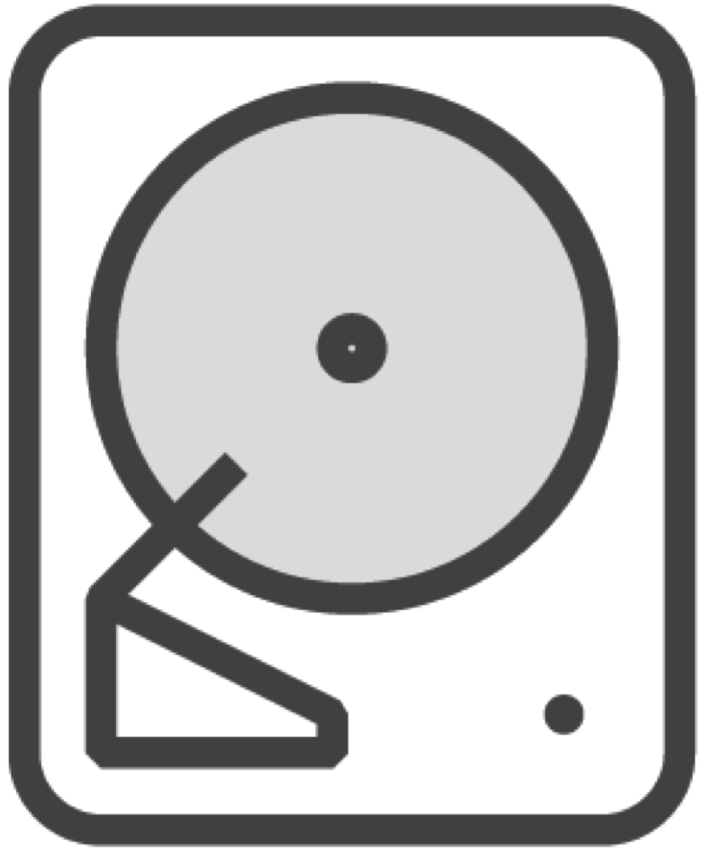


The more CPU time an application gets, the faster it can execute

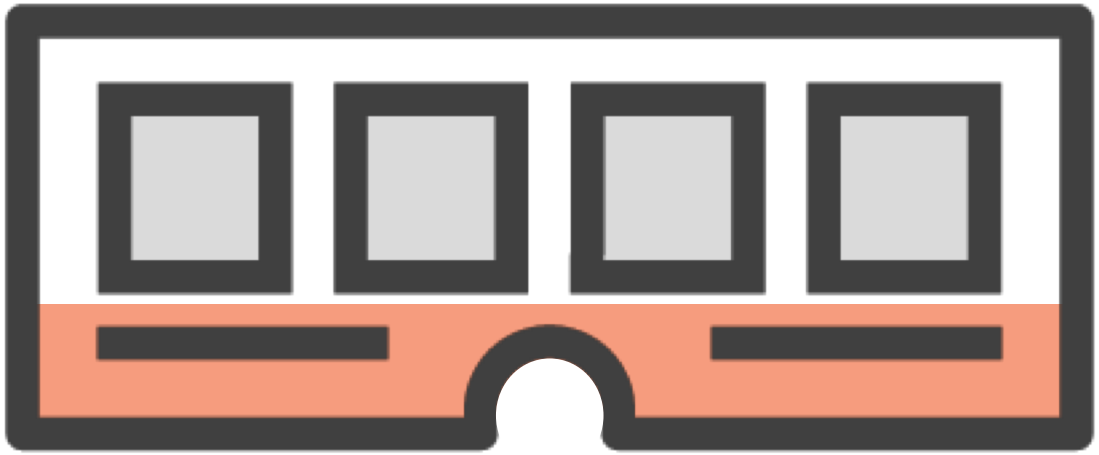
A high run queue
for a prolonged period of time
is a sign of saturation

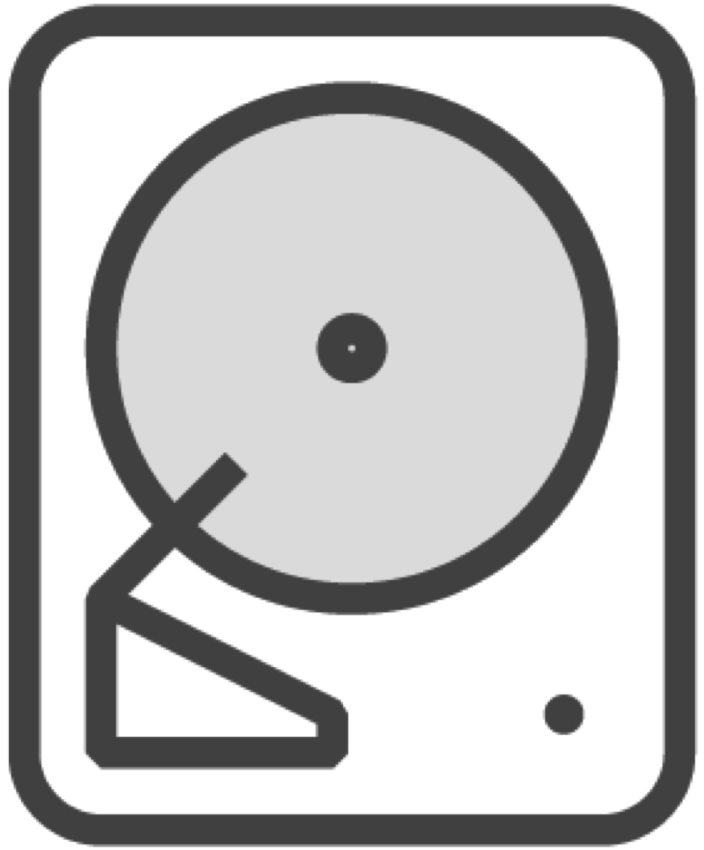
Linux Swapping



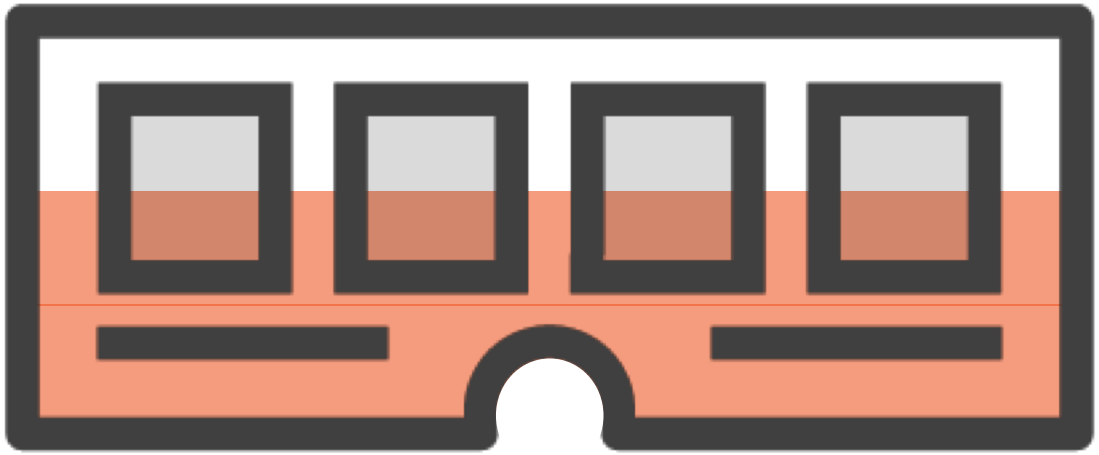


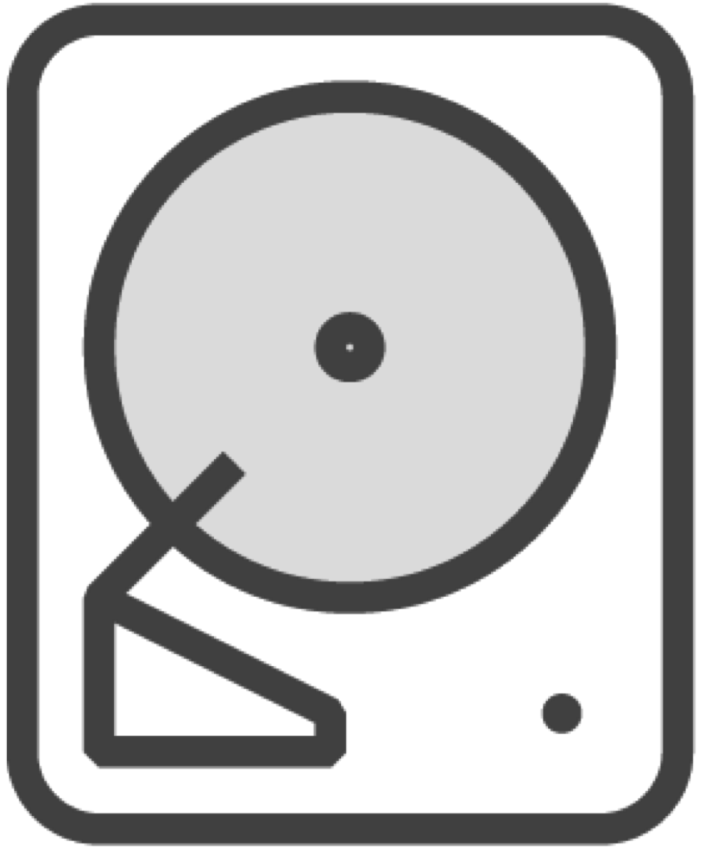
Linux Swapping



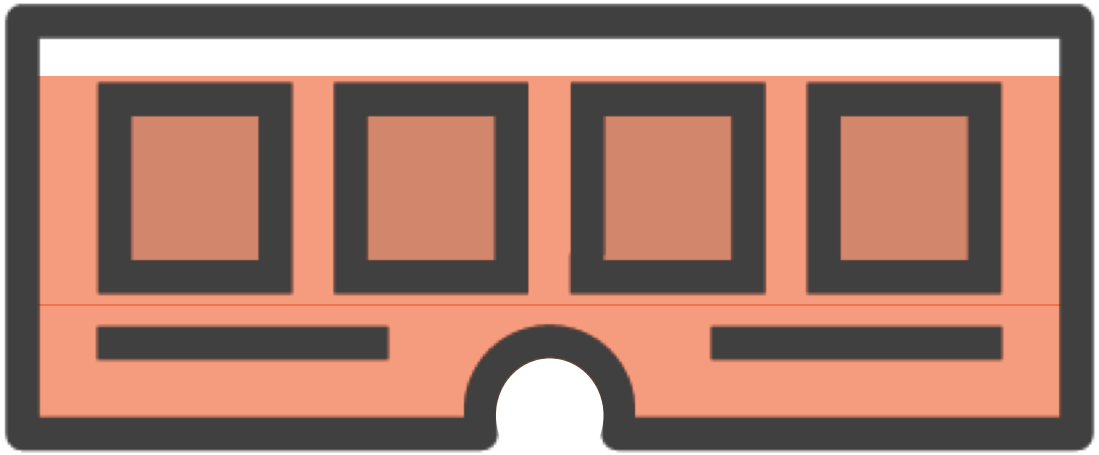


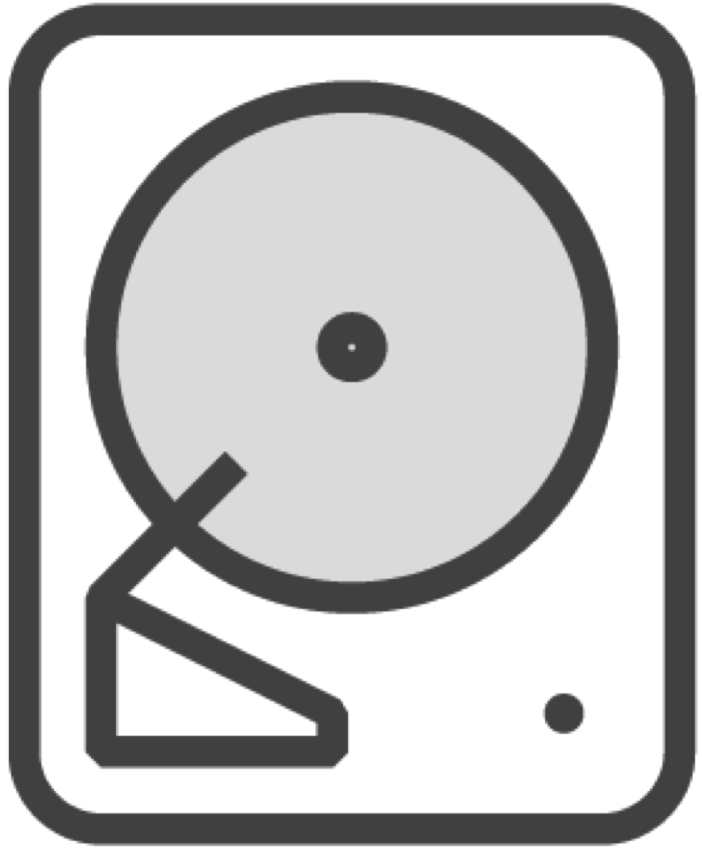
Linux Swapping



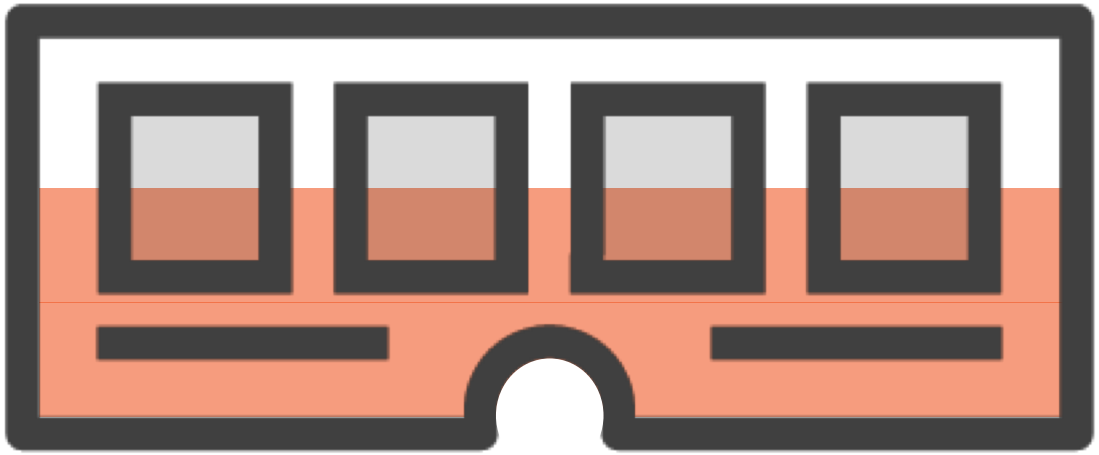


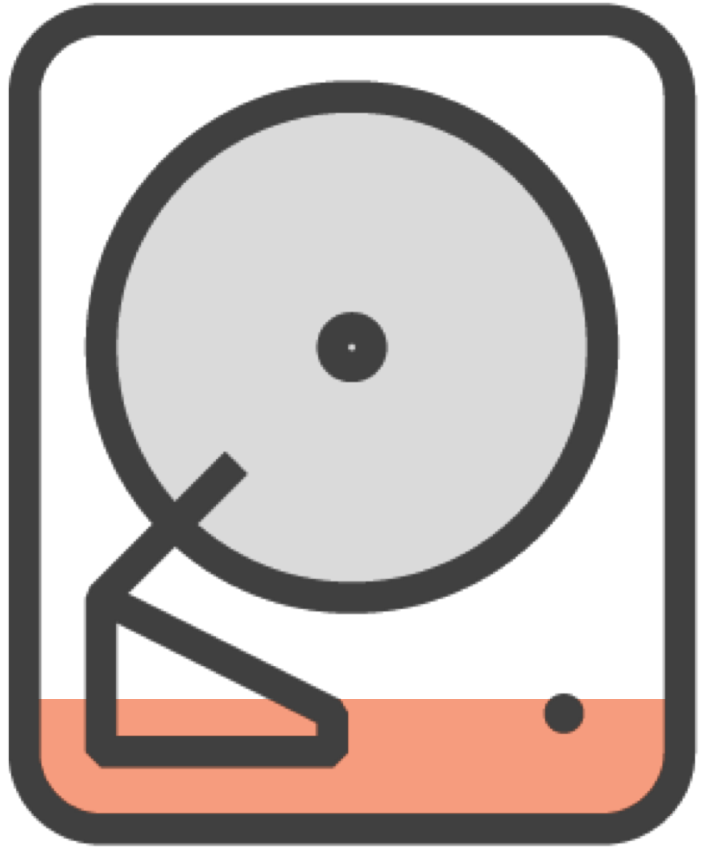
Linux Swapping





Linux Swapping

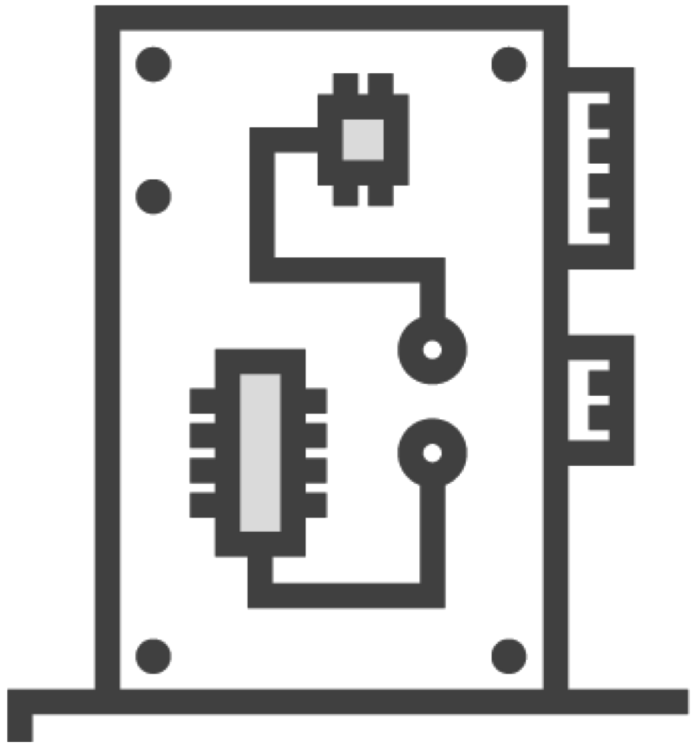




Swap Analysis Caveats

The Linux kernel may choose to swap out hardly used pages even if there is sufficient RAM

To determine memory saturation, combine Swap usage with memory usage



netstat

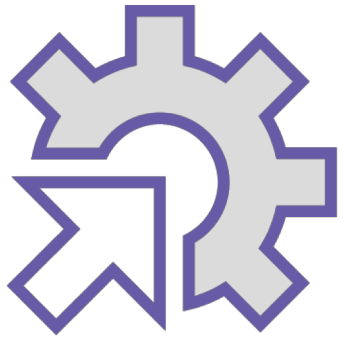
nicstat

typeperf

sar

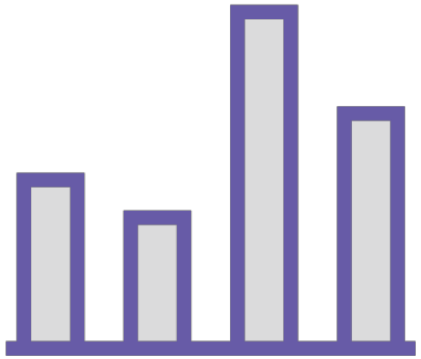
GUI tools

Monitoring Pipeline



Collect

Store



Analyze / Visualize

Application Performance Metrics

Performance Measurements

Latency (Response time)

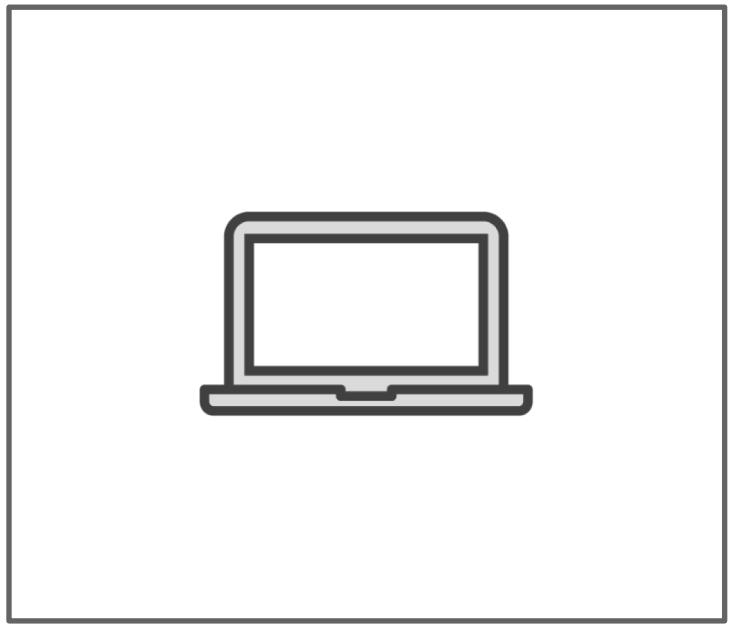
Elapsed time

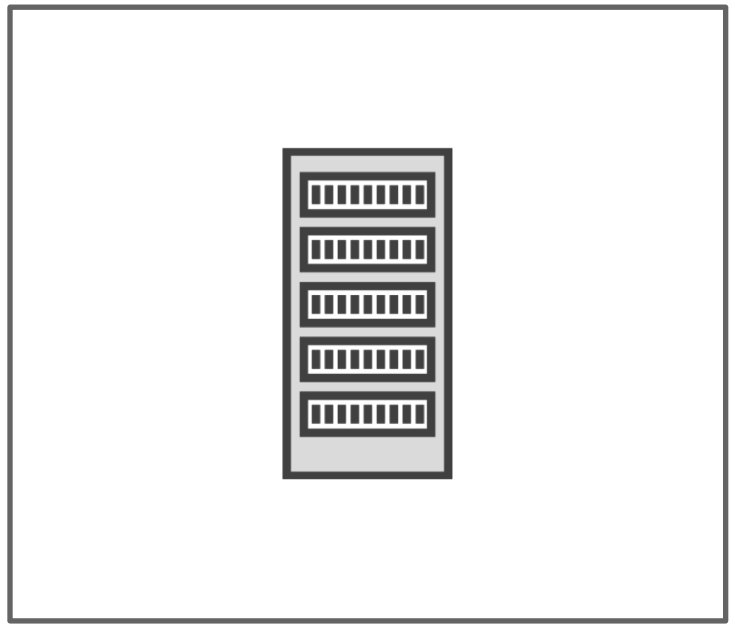
Throughput

Latency

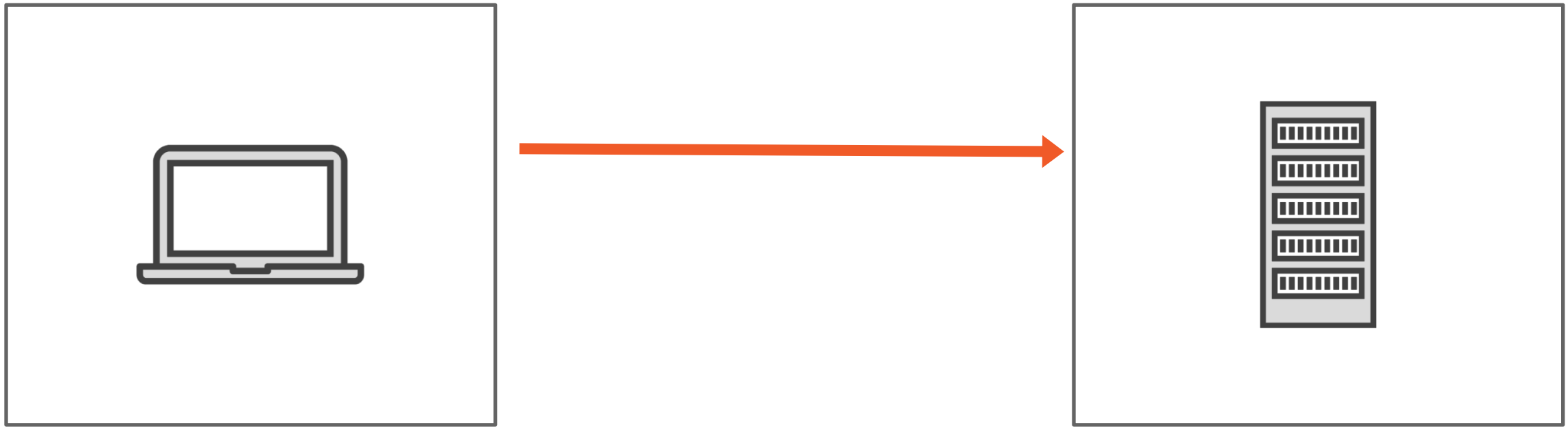
Is the amount of time required to complete a unit of work

Latency



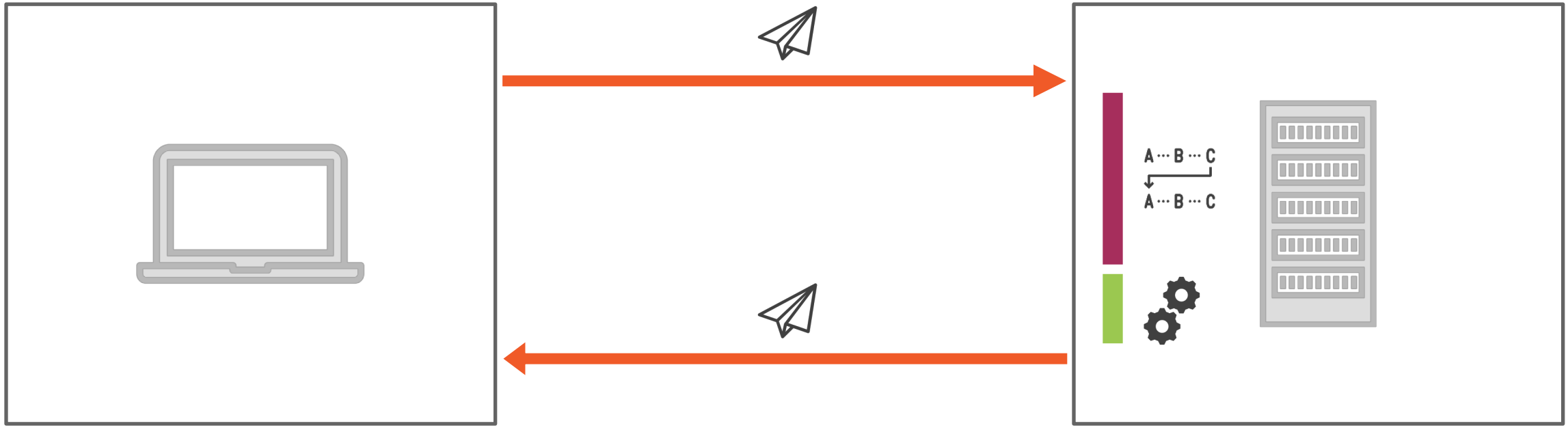


Latency

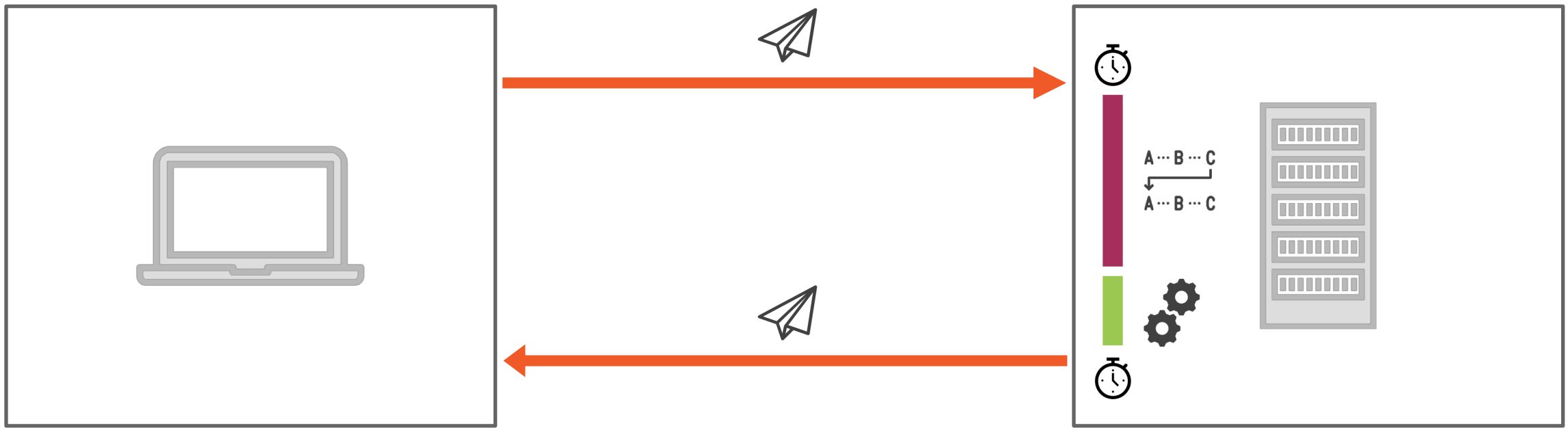


Latency

Latency



Latency



Elapsed Time

Measures the time taken for a batch of operations to complete

Examples of Batch Operations

Writing records to a database in batches of 1000

Calculating the standard deviation of hourly meter readings for a year


measurement cost
operation cost

```
public void doTest(String input) {
    long start = System.currentTimeInMillis();
    result = input.toLowerCase();
    long end = System.currentTimeInMillis();
    System.out.println("Response time: " + (end - start));
}
```

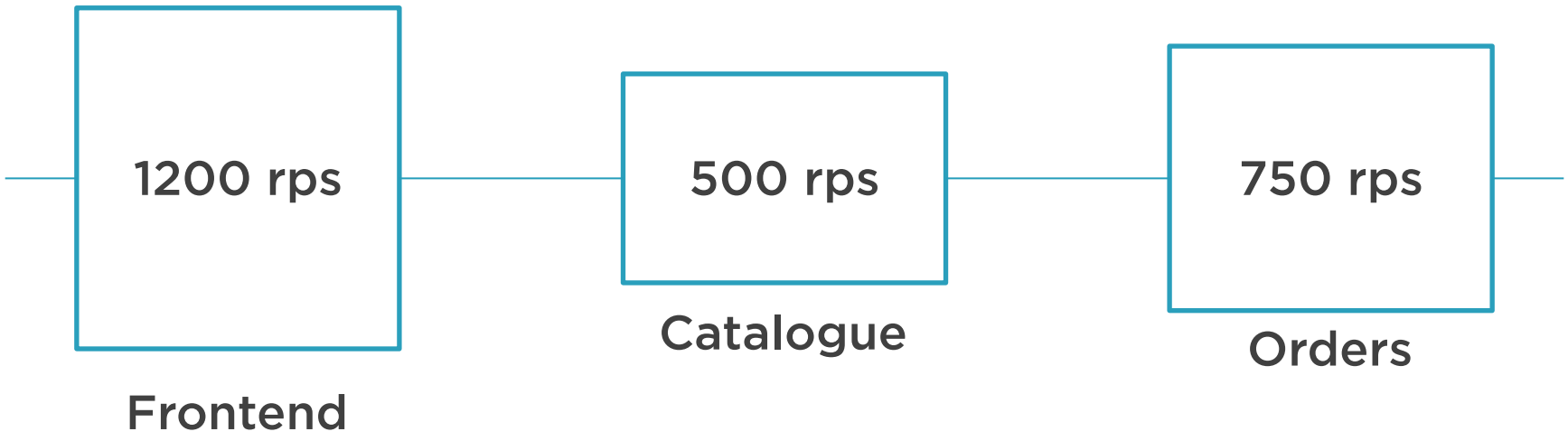
```
public void doTest(String[] inputs) {
    long start = System.currentTimeInMillis();
    for (int i = 0; i < inputs.length; i++) {
       result = input.toLowerCase();
    }
    long end = System.currentTimeInMillis();
    System.out.println("Elapsed time: " + (end - start));
}
```

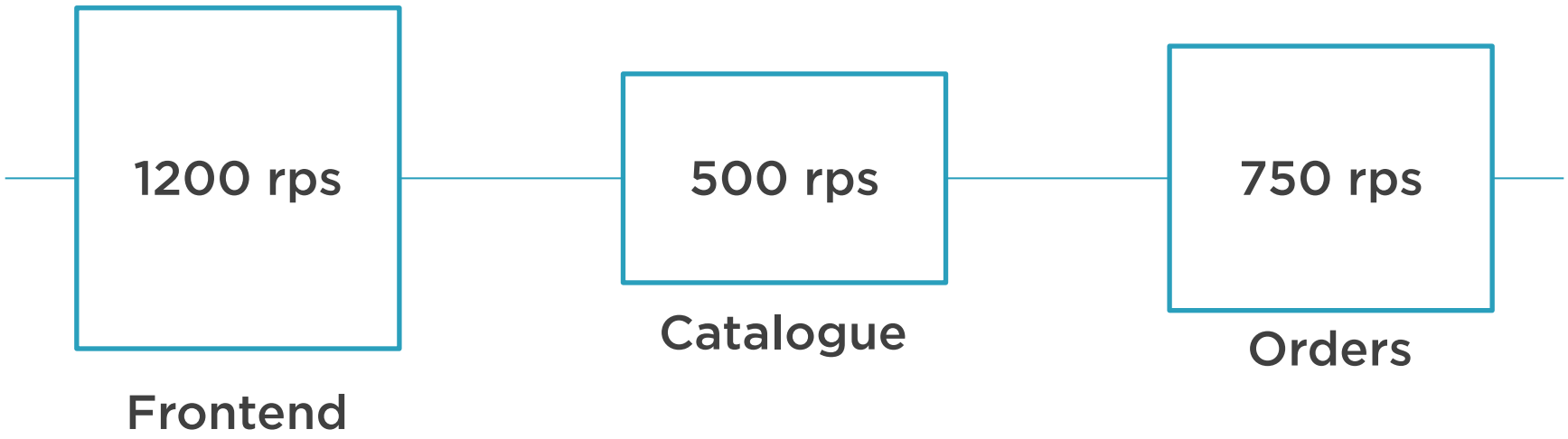
To write accurate microbenchmarks use a library like the Java Microbenchmarking Harness

Throughput

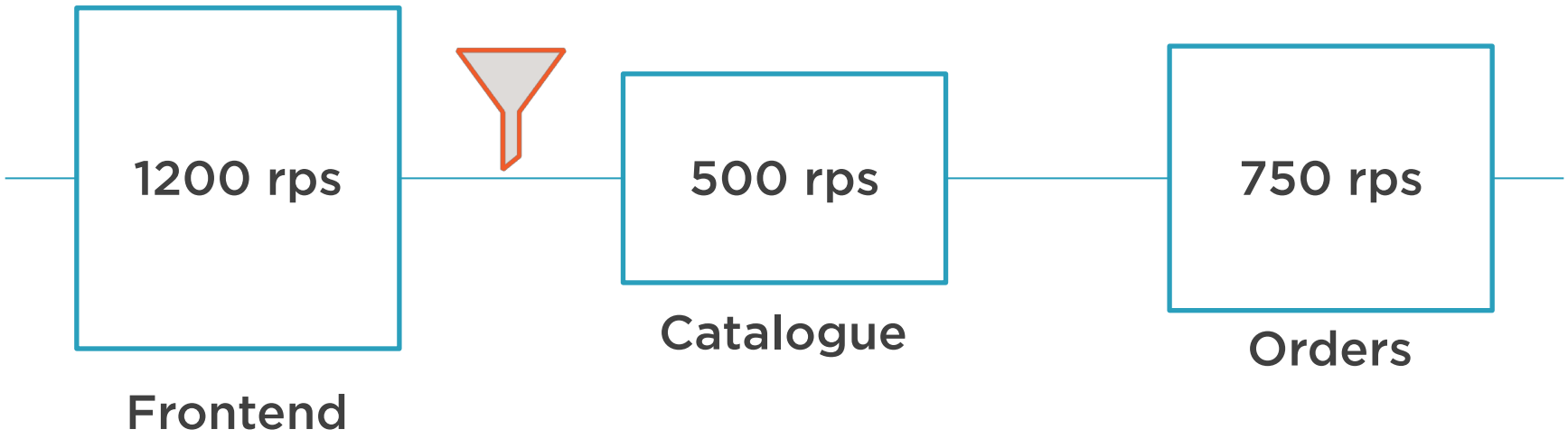
Is the amount of work that an application can accomplish per unit of time

The higher the latency of an application, the higher it's throughput is

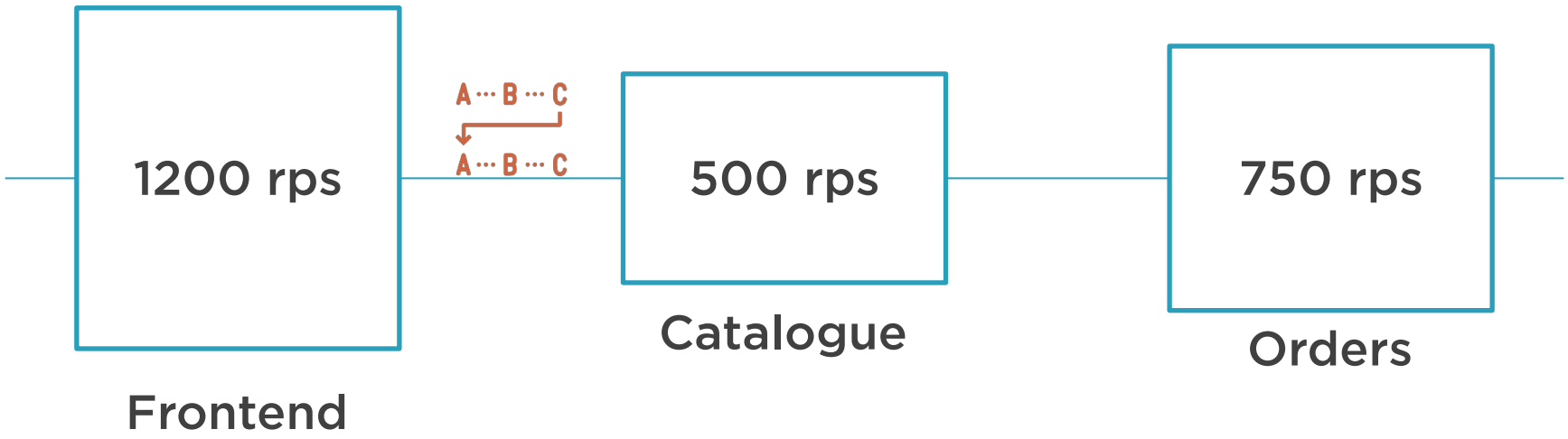

1200 rps
Frontend
500 rps
Catalogue
750 rps
Orders



Effective throughput: 500 rps



Effective throughput: 500 rps



Effective throughput: 500 rps

Measuring Application Performance

Production Monitoring

Performance Testing

Production Monitoring

Operations activity

Performance Testing

Production Monitoring

Operations activity

Collect and aggregate metric data

Performance Testing

Production Monitoring

Operations activity

Collect and aggregate metric data

Metric data storage

Performance Testing

Production Monitoring

Operations activity

Collect and aggregate metric data

Metric data storage

Analysis, visualization, alerting

Performance Testing

Production Monitoring

Operations activity

Collect and aggregate metric data

Metric data storage

Analysis, visualization, alerting

Performance Testing

Development activity

Production Monitoring

Operations activity

Collect and aggregate metric data

Metric data storage

Analysis, visualization, alerting

Performance Testing

Development activity

Define application/component under test

Production Monitoring

Operations activity

Collect and aggregate metric data

Metric data storage

Analysis, visualization, alerting

Performance Testing

Development activity

Define application/component under test

Generate load against application

Production Monitoring

Operations activity

Collect and aggregate metric data

Metric data storage

Analysis, visualization, alerting

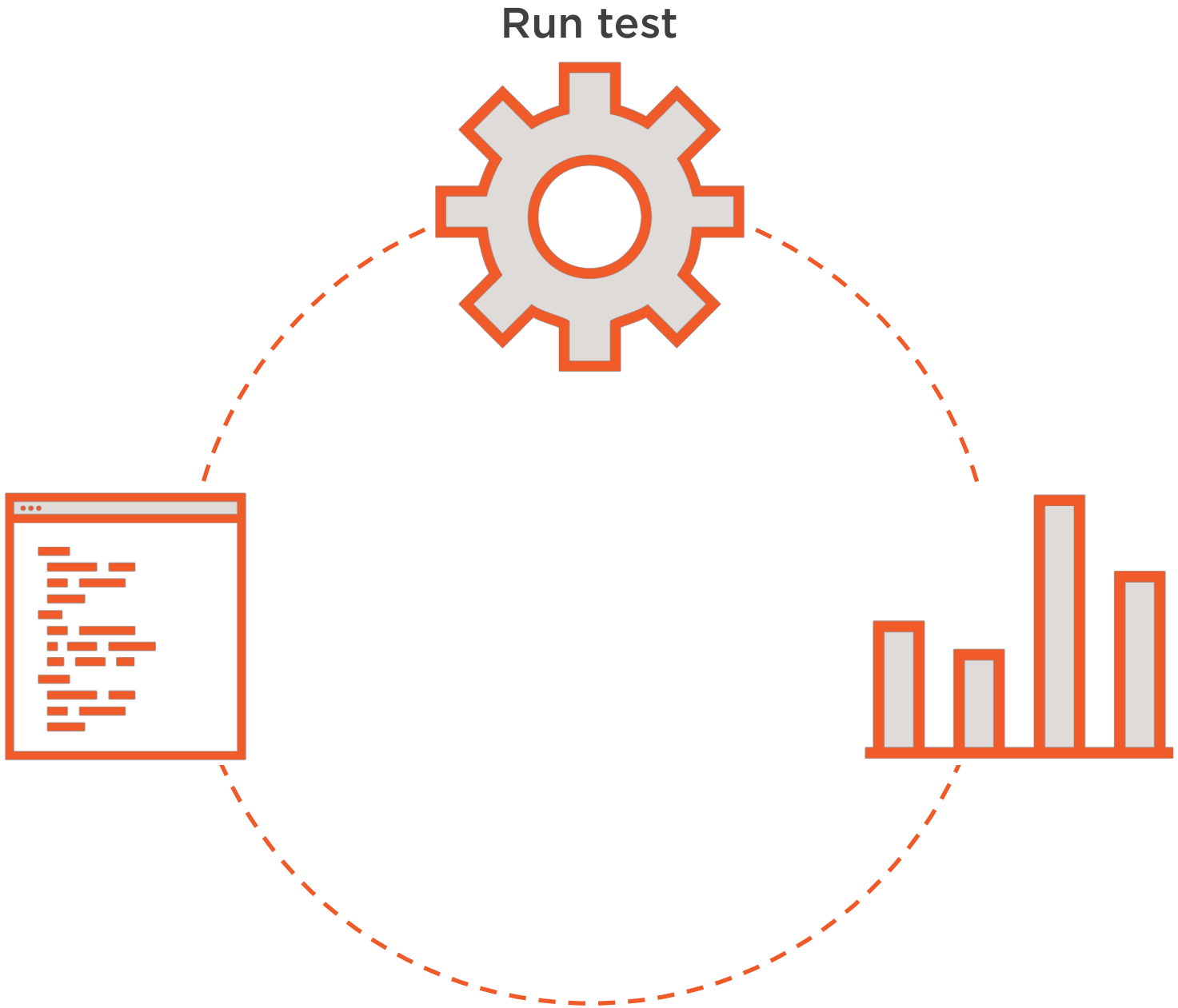
Performance Testing

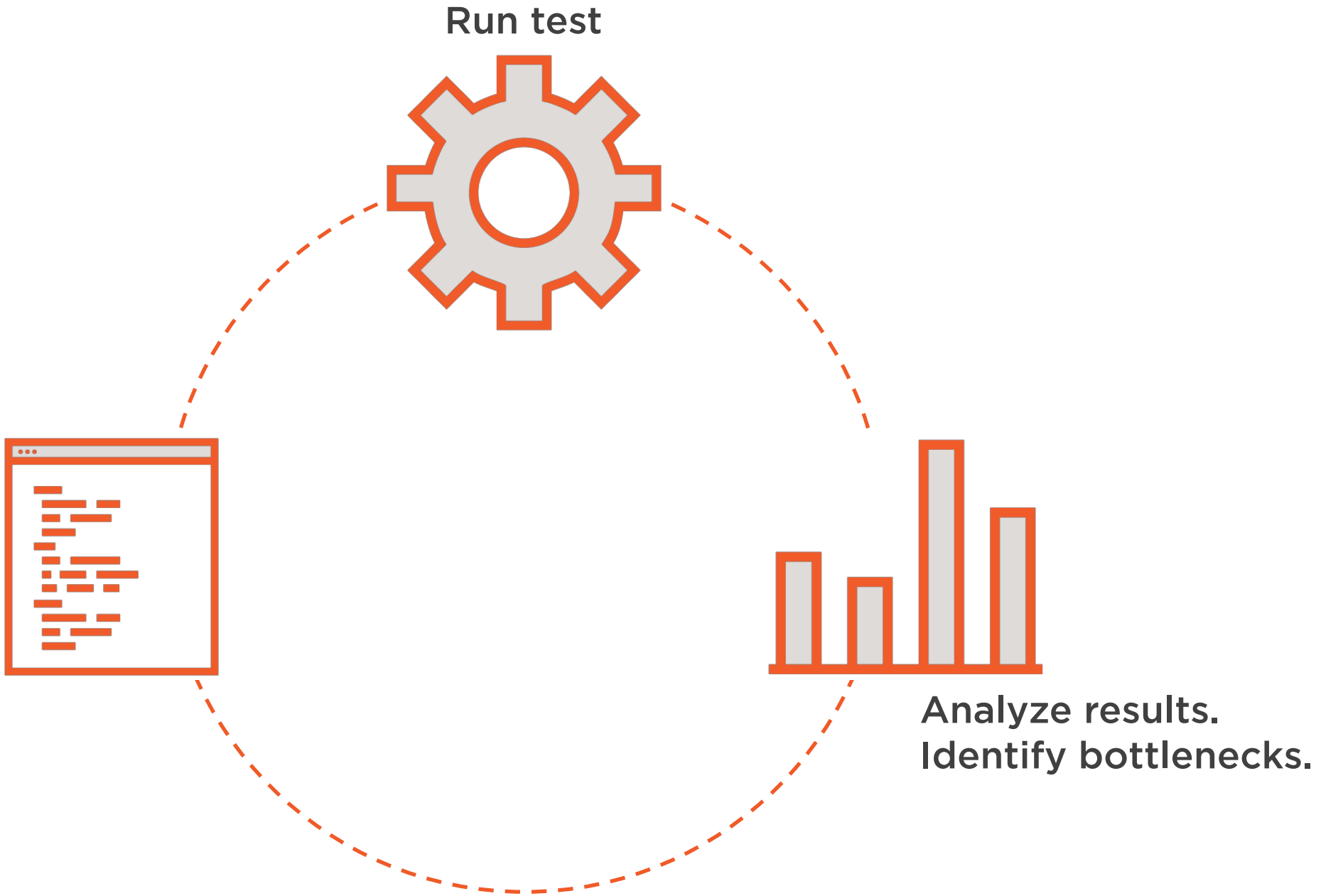
Development activity

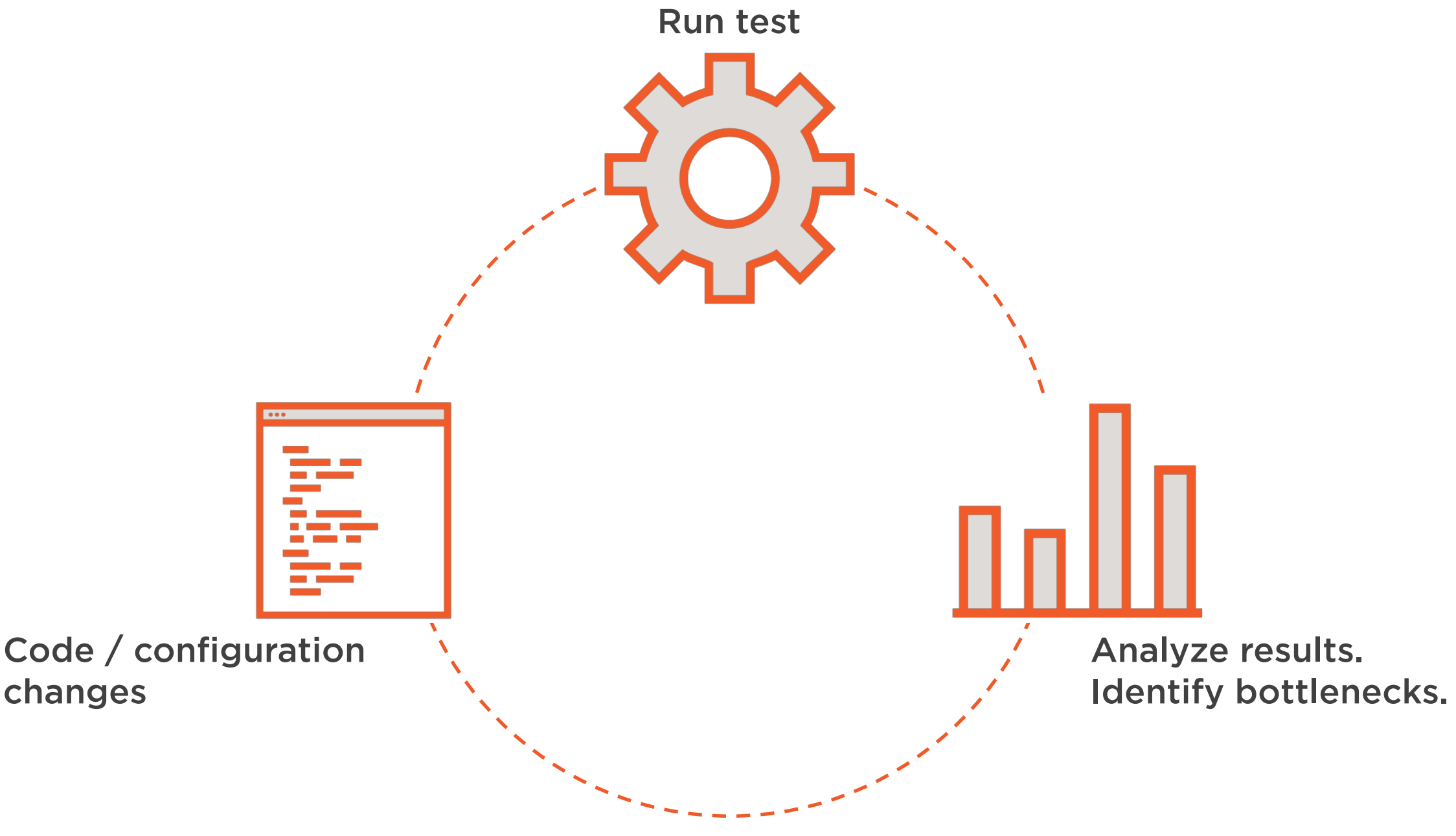
Define application/component under test

Generate load against application

Analyze results


Run test


Run test
Analyze results.
Identify bottlenecks.


Run test
Code / configuration changes
Analyze results.
Identify bottlenecks.

Both Are Needed

Performance testing

Helps to anticipate and solve performance problems

Production monitoring

Gives aggregated information about user's actual experience

Performance Testing Tips

Use representative data

Input data should be representative of how the code will actually be used

Have a warm-up period

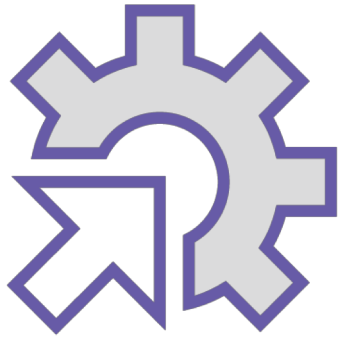
A period of time when the test is running but recording has not yet started

Warm-up Period

Allow JVM to complete:

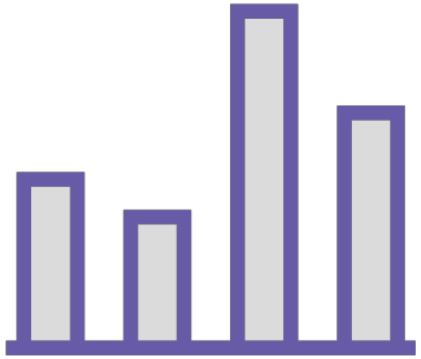
- **Classloading**
- **Runtime optimizations**
- **Cache warming**
- **Code compilation**

Production Monitoring Pipeline



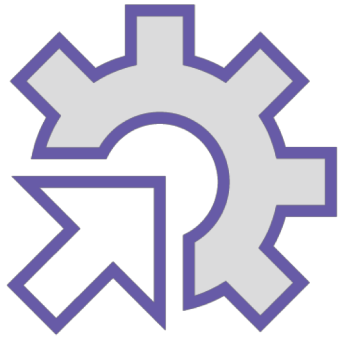
Collect

Store



Analyze / Visualize

Production Monitoring Pipeline



Collect

Dropwizard metrics

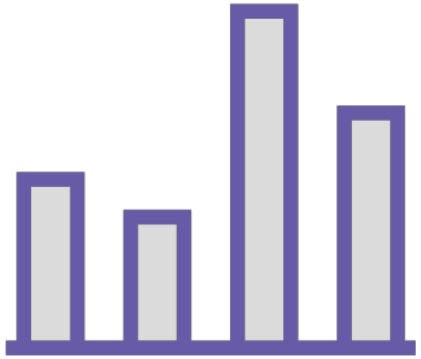
Micrometer metrics

Spectator

OpenCensus

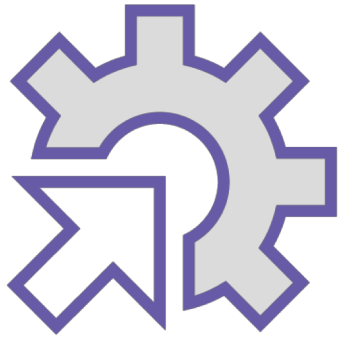
Prometheus Lib

Store



Analyze / Visualize

Production Monitoring Pipeline



Collect

Dropwizard metrics

Micrometer metrics

Spectator

OpenCensus

Prometheus Lib

Store

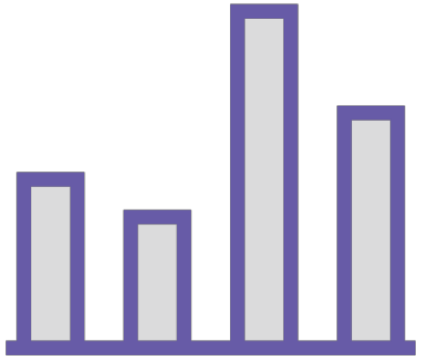
InfluxDB

Elasticsearch

Graphite

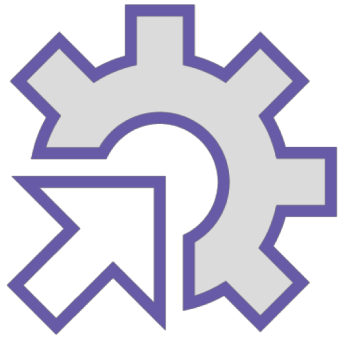
OpenTSDB

Prometheus TSDB



Analyze / Visualize

Production Monitoring Pipeline



Collect

Dropwizard metrics

Micrometer metrics

Spectator

OpenCensus

Prometheus Lib

Store

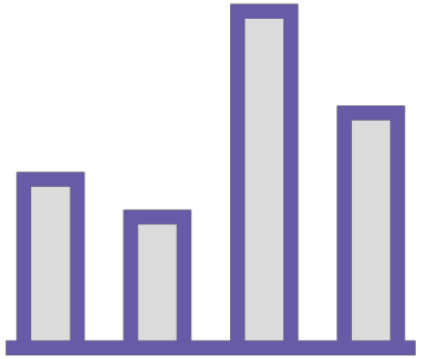
InfluxDB

Elasticsearch

Graphite

OpenTSDB

Prometheus TSDB



Analyze / Visualize

Grafana

Kibana

JMX tools

Prometheus dashboards

```
public void doSomeWork() {
    logger.debug("Starting doSomeWork()");
    long start = System.currentTimeInMillis();
    ...
    long end = System.currentTimeInMillis();
    logger.debug("Completed doSomeWork() in " + (end - start) + " ms");
}
```

Manual Instrumentation

Every suspected code block must be manually instrumented

Might miss other information

Performance Testing Tools - Java

Java Mission Control

Tool for collecting low level and detailed runtime information from a JVM

Integrates:

- jstat (JVM statistics)
- jinfo (JVM configuration info)
- jmap (Java memory map)
- jstack (stack traces)
- jconsole (JMX MBeans)
- Java Flight Recorder UI

```
java -Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=<your desired port
number> -Djava.rmi.server.hostname=<your IP address> -
Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

Enabling JMX Remote Monitoring

```
java -Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=<your desired port
number> -Djava.rmi.server.hostname=<your IP address> -
Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

Securing JMX Remote Monitoring

- Pluggable login modules for authentication
- By default; password and certificate based login

Summary

How hardware affects performance

How to measure system performance

Application performance metrics

Performance testing tool suite