

# **8 PERFORMANCE TIPS FOR MARATHON**

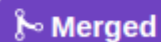
# TOMASZ JANISZEWSKI

 JANISZT

 JANISZ

allegro Tech

## Add Allegro Group to list of companies using marathon #1442



**drexin** merged 1 commit into `mesosphere:master` from `unknown repository` on Apr 30, 2015

# **#1: ENABLE METRICS**

We collect a lot of metrics in Marathon. You can collect those metrics via the `/metrics` endpoint. With those reporters you can transfer the data into either Graphite or DataDog and see the values over time.

— Release Notes: v0.13.0



StatsD



```
--reporter_graphite=tcp://localhost:2003?prefix=marathon-test&interval=10
```



sun.management.ThreadImpl.get..  
sun.management.ThreadImpl.get..  
com.codahale.metrics.jvm.Thre..  
com.codahale.metrics.jvm.Thre..  
com.codahale.metrics.jvm.Thre..  
com.codahale.metrics.jvm.Thre..  
com.codahale.metrics.graphite..  
com.codahale.metrics.graphite..  
com.codahale.metrics.Scheduled..  
com.codahale.metrics.Scheduled..  
java.util.concurrent.Executors..  
java.util.concurrent.FutureTas..  
java.util.concurrent.Scheduled..  
java.util.concurrent.Scheduled..  
java.util.concurrent.ThreadPoo..  
java.util.concurrent.ThreadPoo..  
E.. java.lang.Thread.run:745  
T.. Thread 140304099430904

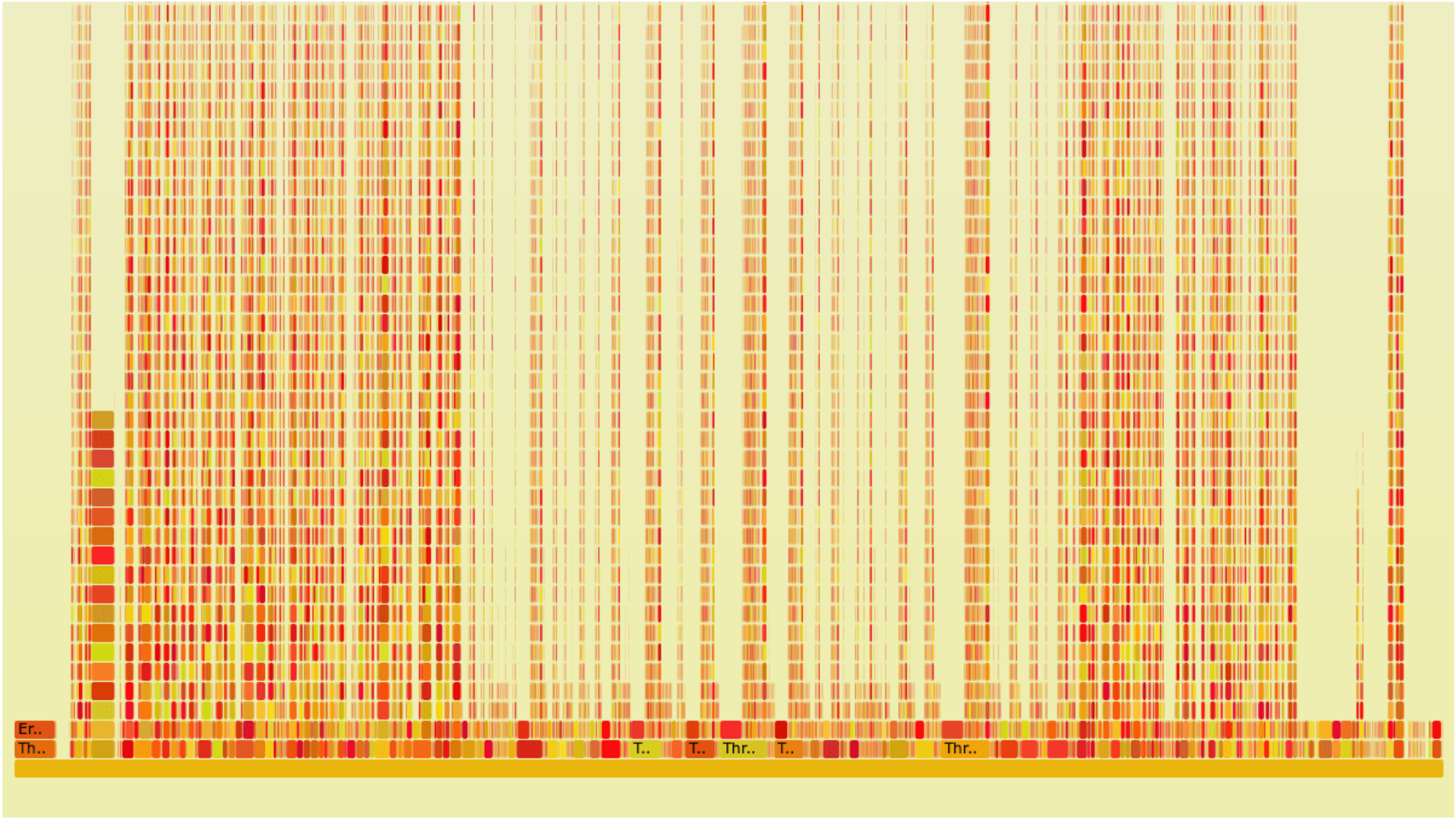
Thr..

Th..

Th..

T..

```
--reporter_graphite=tcp://localhost:2003?prefix=marathon-test&interval=55
```







**kamon**

## Update DropWizard #5495



Open

janisz wants to merge 1 commit into `mesosphere:releases/1.3` from `janisz:release/1.3/update_dropwizard`

**#2: TUNE JVM**

GC



```
-XX:+PrintTenuringDistribution  
-XX:+PrintGC  
-XX:+PrintGCDetails  
-XX:+PrintGCTimeStamps  
-XX:+PrintGCApplicationStoppedTime  
-Xloggc:/var/log/marathon/gc.log  
-XX:+UseGCLogFileRotation  
-XX:NumberOfGCLogFiles=2  
-XX:GCLogFileSize=50M
```

## FLAGS

- ✓ PrintGCDetails Flag
- ✓ PrintTenuringDistribution Flag
- ✓ PrintGCCause Flag
- ✓ PrintGCApplicationStoppedTime Flag

## HEAP USAGE

- ✓ Metaspace Threshold Triggered GC
- ✗ Heap Too Small Indicators
- ✗ Premature Promotion

## PAUSE TIME

- ✗ High Pause Times
- ✓ Application Throughput

## SYSTEM GC

- ✓ Calls to System.gc()

## MIXED COLLECTIONS

- i Mixed Collection Counts

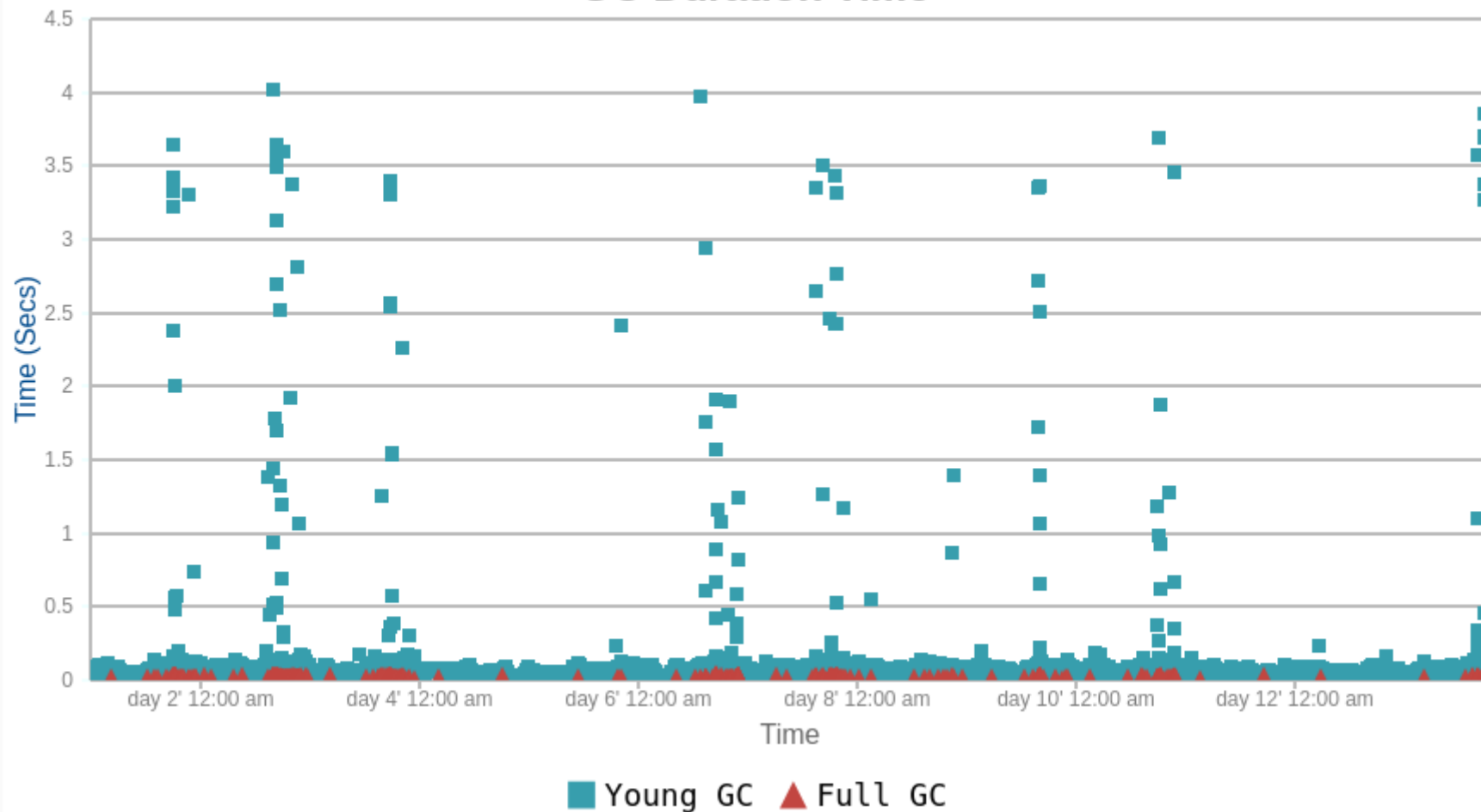
## CPU USAGE

- ✗ High kernel times

## LOG INFORMATION

- ✓ Log duration 12 days 18 hours 33 minutes

## GC Duration Time



**AKKA**

Throughput defines the maximum number of messages to be processed per actor before the thread jumps to the next actor. Set to 1 for as fair as possible.

```
-Dakka.default-dispatcher.throughput=20
```

# **#3: OPTIMIZE ZOOKEEPER**

**SIZE**



```
422 - Failed to deploy app [/really/important/app] to [prod].
Caused by: (http status: 422 from https://production).
RESPONSE: [{
  "message": "Object is not valid",
  "details": [{
    "path": "/",
    "errors": [
      "The way we persist data in ZooKeeper would exceed the maximum ZK node
      size (1024000 bytes). You can adjust this value via --zk_max_node_size,
      but make sure this value is compatible with your ZooKeeper ensemble!
      See: http://zookeeper.apache.org/doc/r3.3.1/zookeeperAdmin.html#Unsafe+Opt
    ]
  }]
}]
```

```
--zk_max_node_size=2024000
```

```
-Djute.maxbuffer=2024000
```

jute.maxbuffer: If this option is changed, the system property must be set on all servers and clients otherwise problems will arise. This is really a sanity check. ZooKeeper is designed to store data on the order of kilobytes in size.

*ZooKeeper is designed to store data on the order of kilobytes in size [in a single node].*

*“In version 1.4, Marathon stores a group only with references in order to keep node size under 1 MB.”*



— Introducing Marathon 1.4, Joerg Schad, March 29, 2017



Marathon / MARATHON-7167

## Marathon unable to recover from ZK outage

### Details

Type:	 Bug	Status:	<a href="#">OPEN</a>
Priority:	 Medium	Resolution:	Unresolved
Affects Version/s:	Marathon 1.4.1, ... <a href="#">(1)</a>	Fix Version/s:	None
Component/s:	<a href="#">Leader Election</a>		
Labels:	<a href="#">marathon-on-call</a>		
Epic Link:	<a href="#">Marathon Technical Debt</a>		

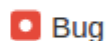


[Marathon](#) / [MARATHON-7656](#)

## Marathon is sensitive to ZK restarts

### Details

Type:



Bug

Status:

OPEN

Priority:



Medium

Resolution:

Unresolved

Affects Version/s:

Marathon 1.3.12,

Fix Version/s:

None

Component/s:

[Leader Election](#)

Labels:

[on-call](#)

[triage](#)



**LATENCY**

**#4: UPDATE TO  
1.3.13\***

# THREADS

```
// src/main/scala/mesosphere/marathon/Main.scala  
setIfNotDefined("scala.concurrent.context.minThreads", "5")  
setIfNotDefined("scala.concurrent.context.numThreads", "x2")  
setIfNotDefined("scala.concurrent.context.maxThreads", "64")
```

```
// src/main/scala/mesosphere/util/ThreadPoolContext.scala  
private val numberOfThreads: Int = System  
    .getProperty("numberOfIoThreads", "100").toInt
```

$$64 + 100 = \text{????}$$

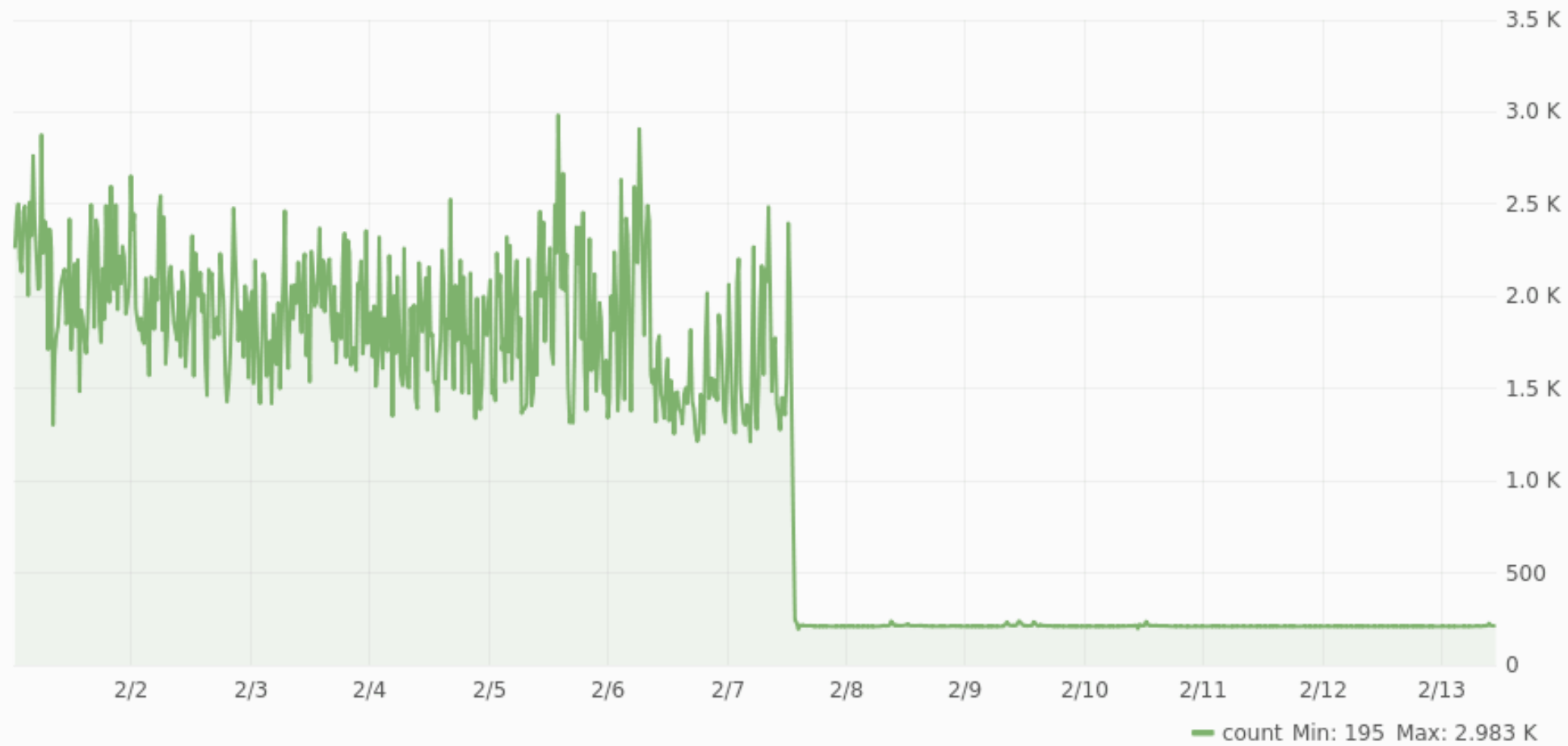
$$64 + 100 \approx 2000$$

$$64 + 100 = O(\text{TASKS})$$



Whenever blocking request is called by the actor new thread is spawned.

## Threads





**Jonathan C Dickinson**

@jcdickinson



Obserwuj

W odpowiedzi do @leppie

[@leppie](#) [@tk\\_riple](#) NOOP optimization: the fastest code you can possibly write is no code at all.

09:47 - 28.10.2015



1



2

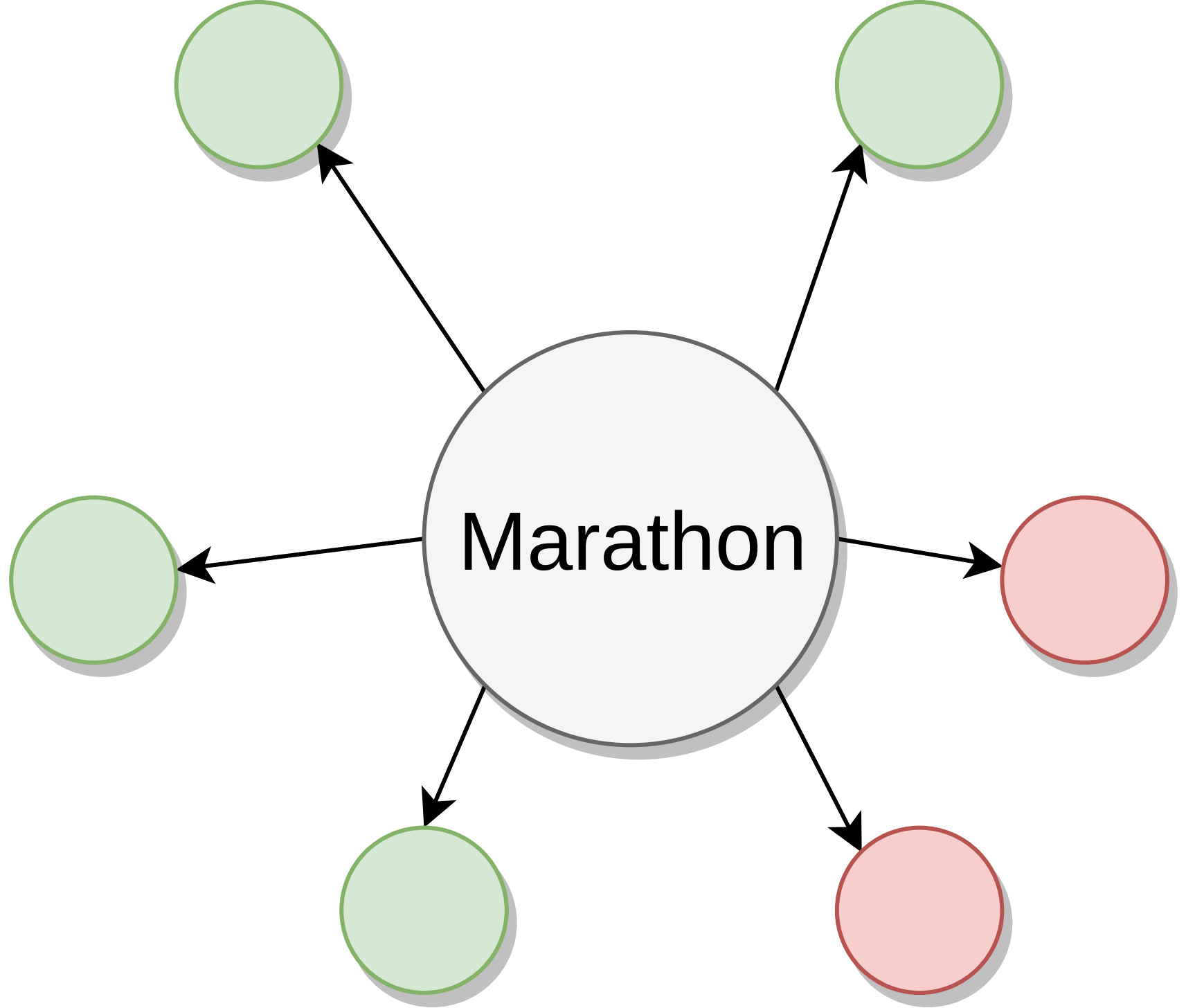


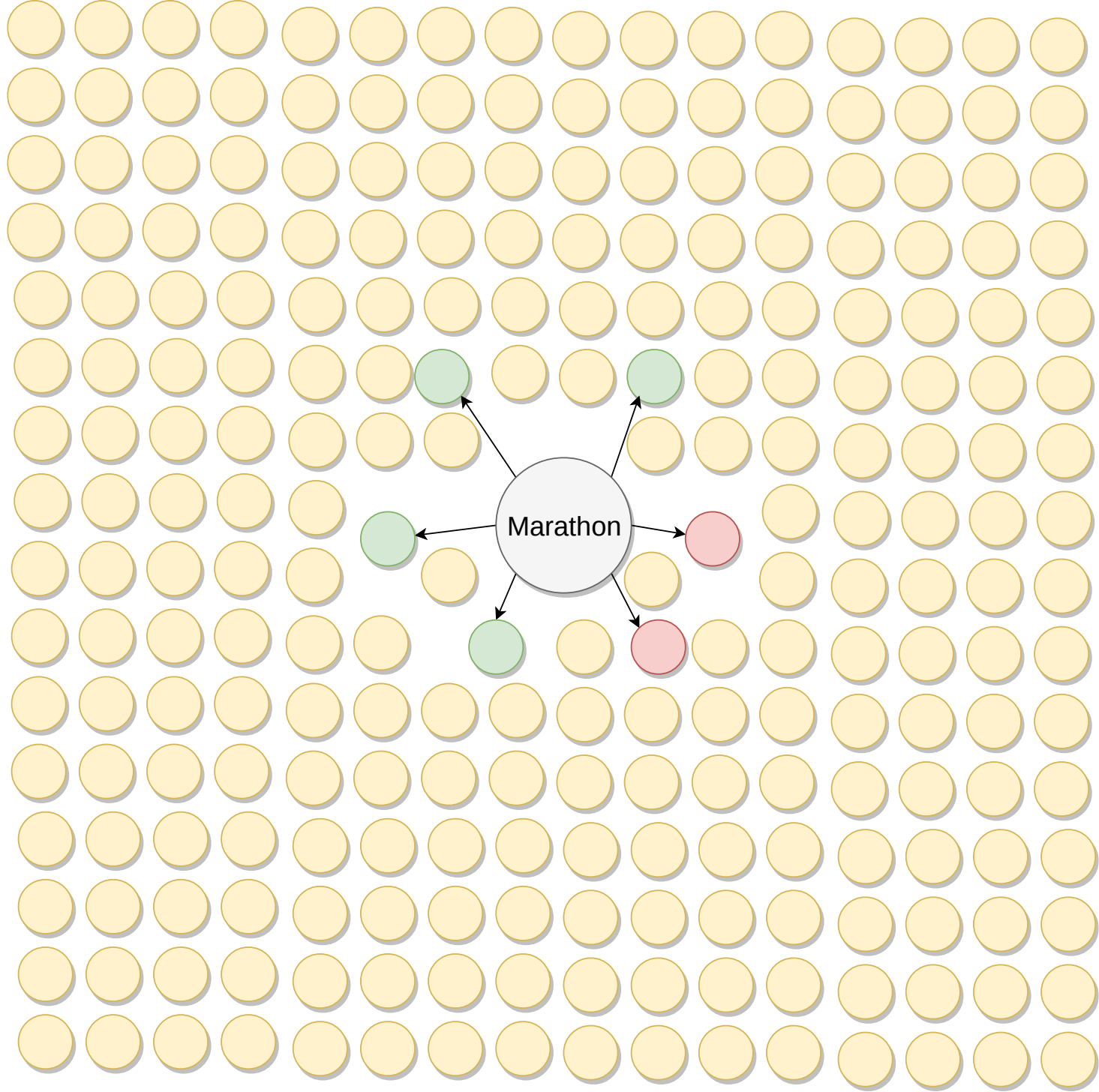
3

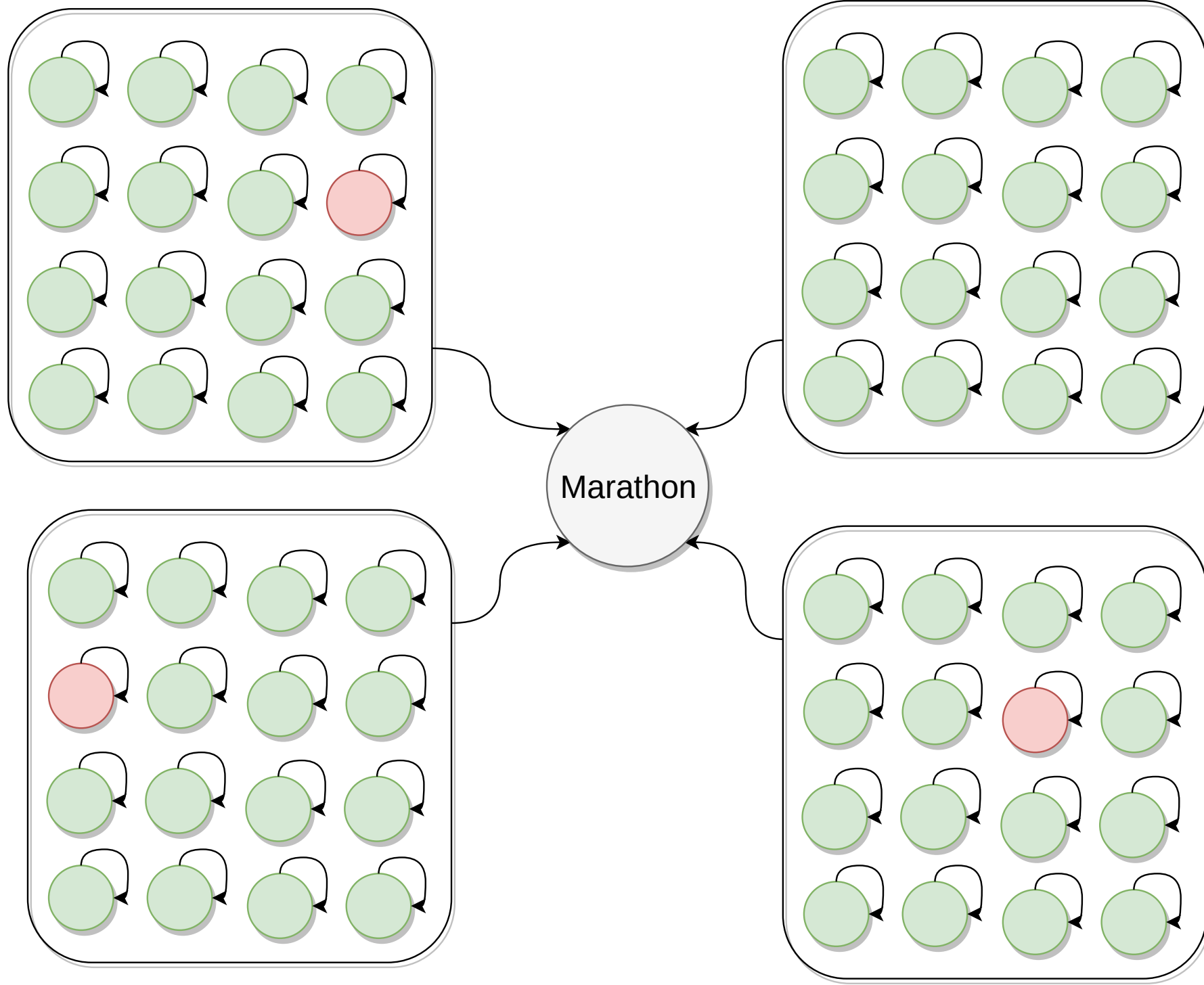


*NOOP optimization: the fastest operation you can possibly perform is no operation at all.*

**HEALTHCHECKS**









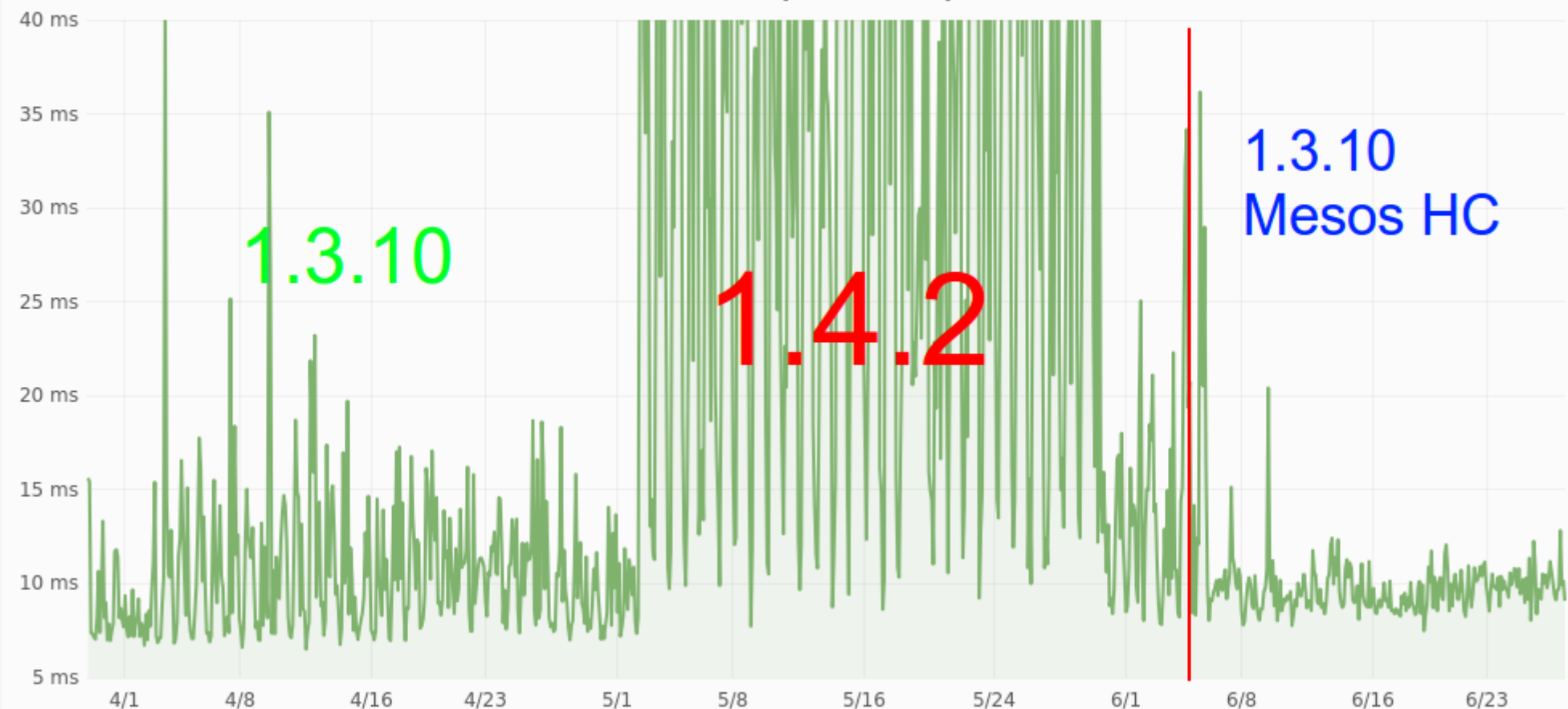
*“Marathon could handle only up to  
3700 tasks with Marathon TCP health  
checks”*

— *Introducing Mesos-native health checks in Apache Mesos – Part II, Gaston Kleiman, January 17, 2017*

*“when using Marathon HTTP checks, we found out that Marathon started to report health failures and return errors to the API requests when the cluster reached the 1900 tasks mark”*

— Introducing Mesos-native health checks in Apache Mesos – Part II, Gaston Kleiman, January 17, 2017

Marathon response times p99







Marathon / MARATHON-7358

## Deployments performance degradation in 1.4

### Details

Type:	 Bug	Status:	IN PROGRESS
Priority:	 High		(View Workflow)
Affects Version/s:	Marathon 1.4.0, ... (3)	Resolution:	Unresolved
Component/s:	Deployments	Fix Version/s:	Marathon 1.4.8
Labels:	<div>bulk-drop-1.10-20170809</div> <div>marathon-triage</div> <div>performance</div> <div>post-1.10-beta1</div>		
Epic Link:	Marathon performance and stability improvements		
Sprint:	Marathon Sprint 1.10-5, Marathon Sprint 1.10-6, Marathon Sprint 1.10-7, Marathon Sprint 1.10-9, Marathon Sprint 1.10-10, Marathon Sprint 1.11-1, Marathon Sprint 1.11-2		
Story Points:	3		
Github Link:	<a href="https://github.com/mesosphere/marathon/pull/5406">https://github.com/mesosphere/marathon/pull/5406</a>		

**#5: DO NOT USE THE  
EVENT BUS**

# CALLBACK

- sends every event
- some events are big ( $\geq 10$  MB)
- deprecated

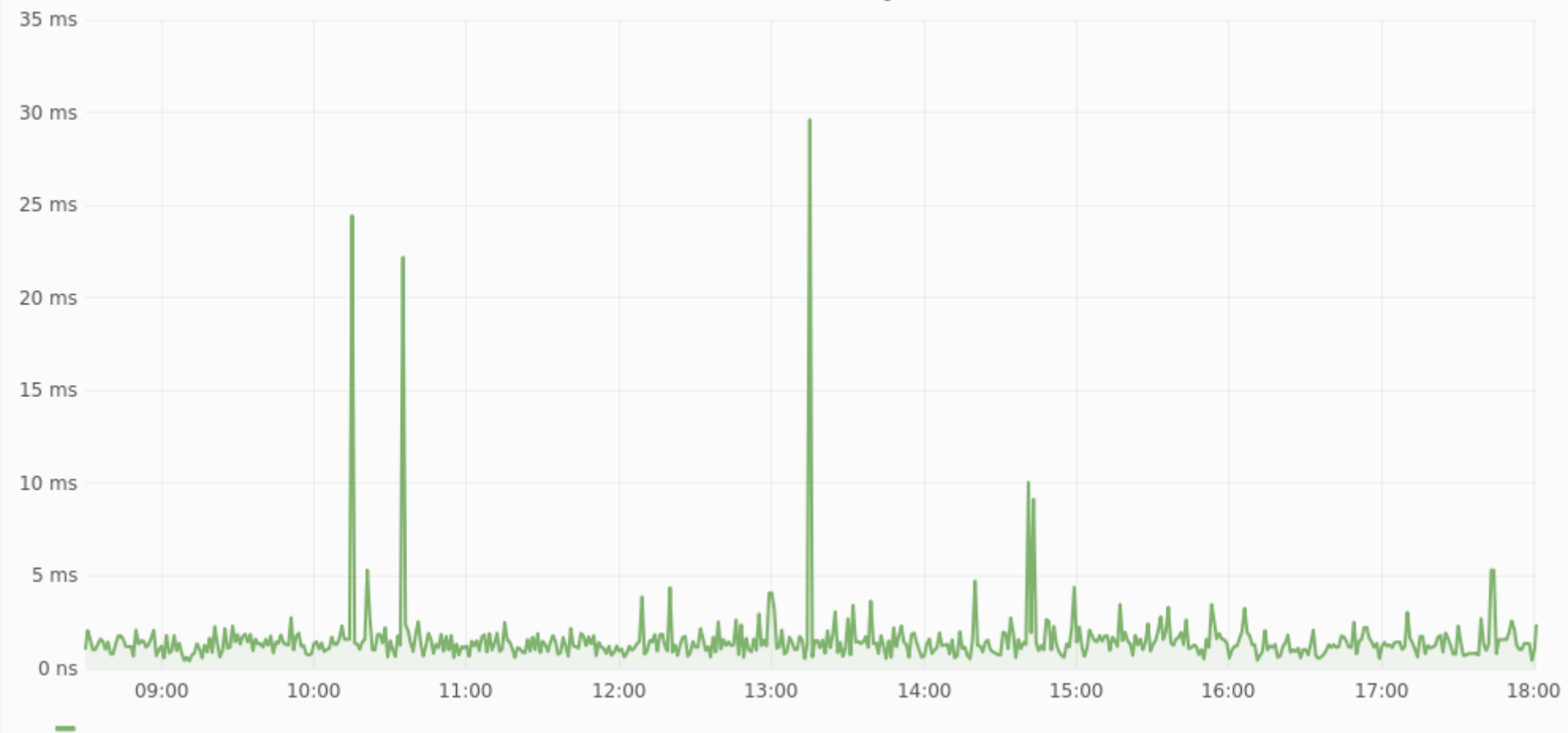
# SSE

- support event filtering (1.3.7)
- some events are big ( $\geq 10$  MB)

Marathon Events delay



**Marathon Events delay (SSE)**



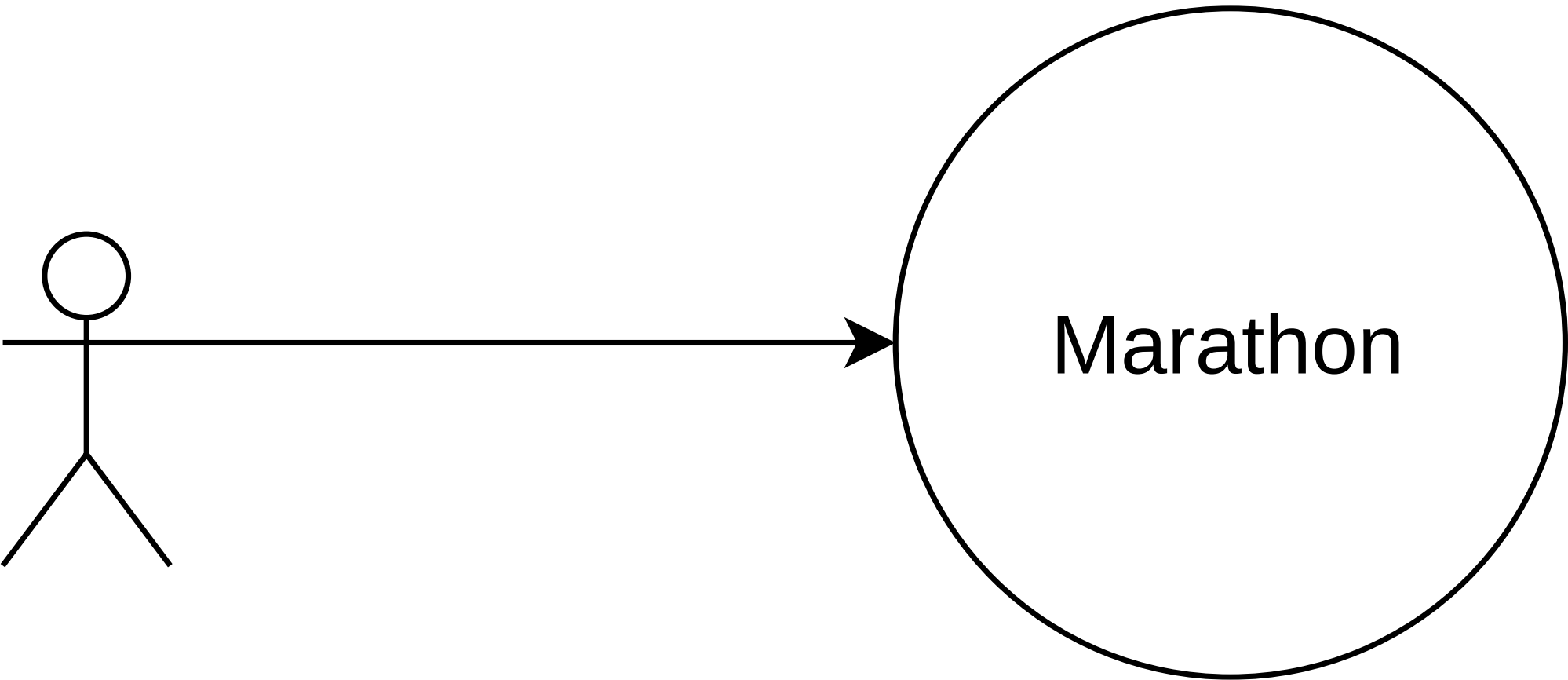


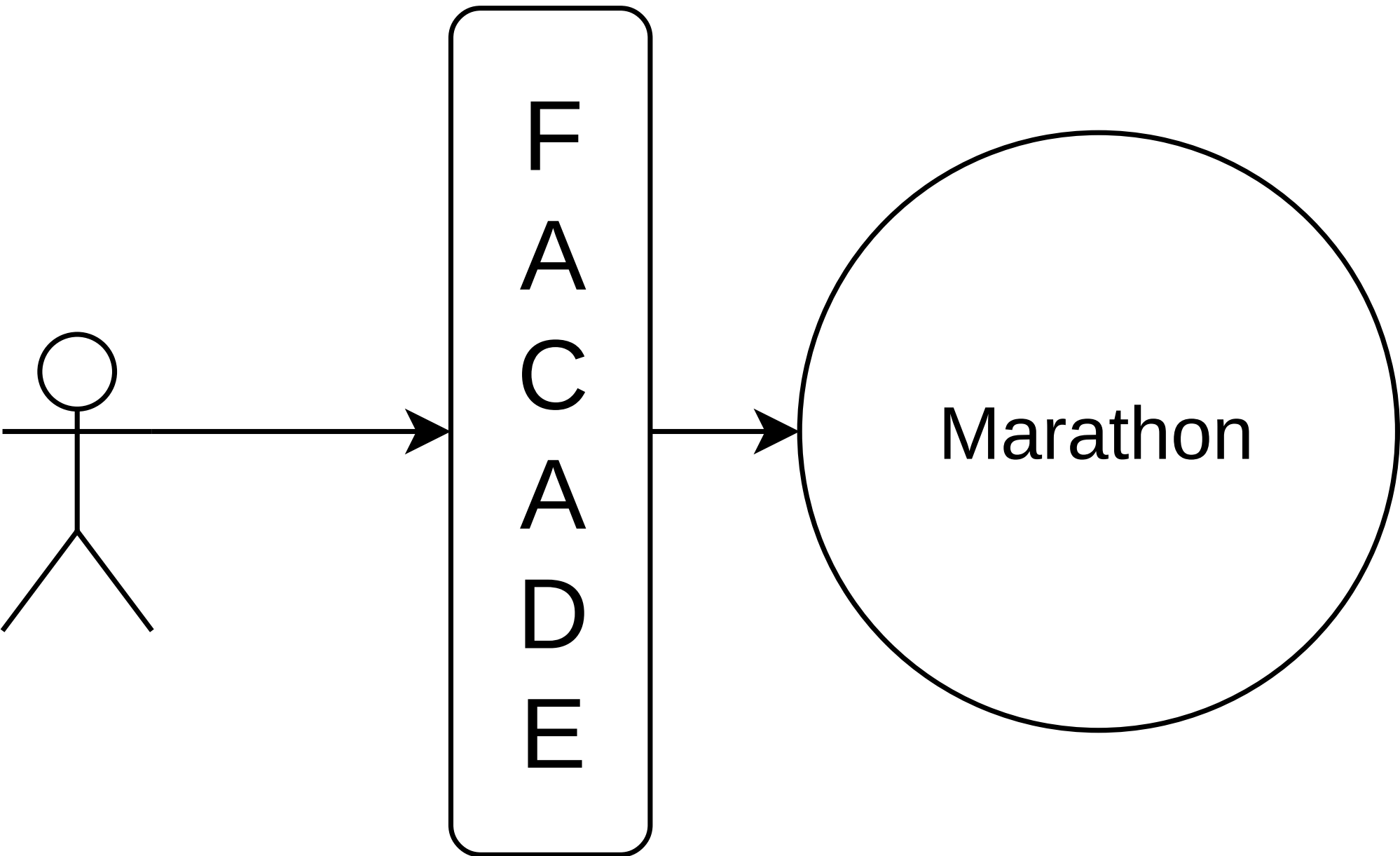
# **#6: USE A CUSTOM EXECUTOR**

# **TASK STATE CHANGE:**

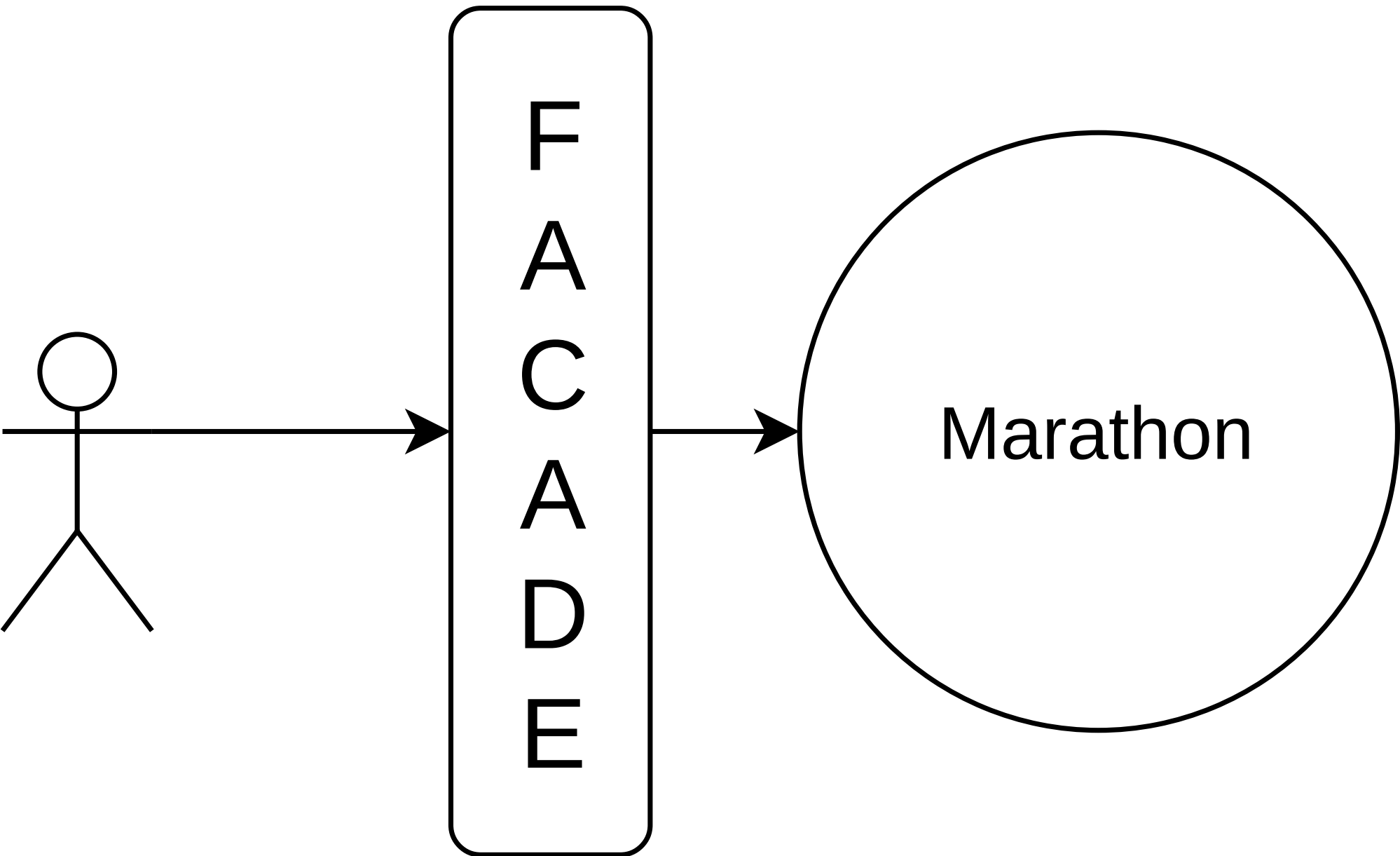
- **STARTING**
- **RUNNING**
- **KILLING**
- **HEALTH**

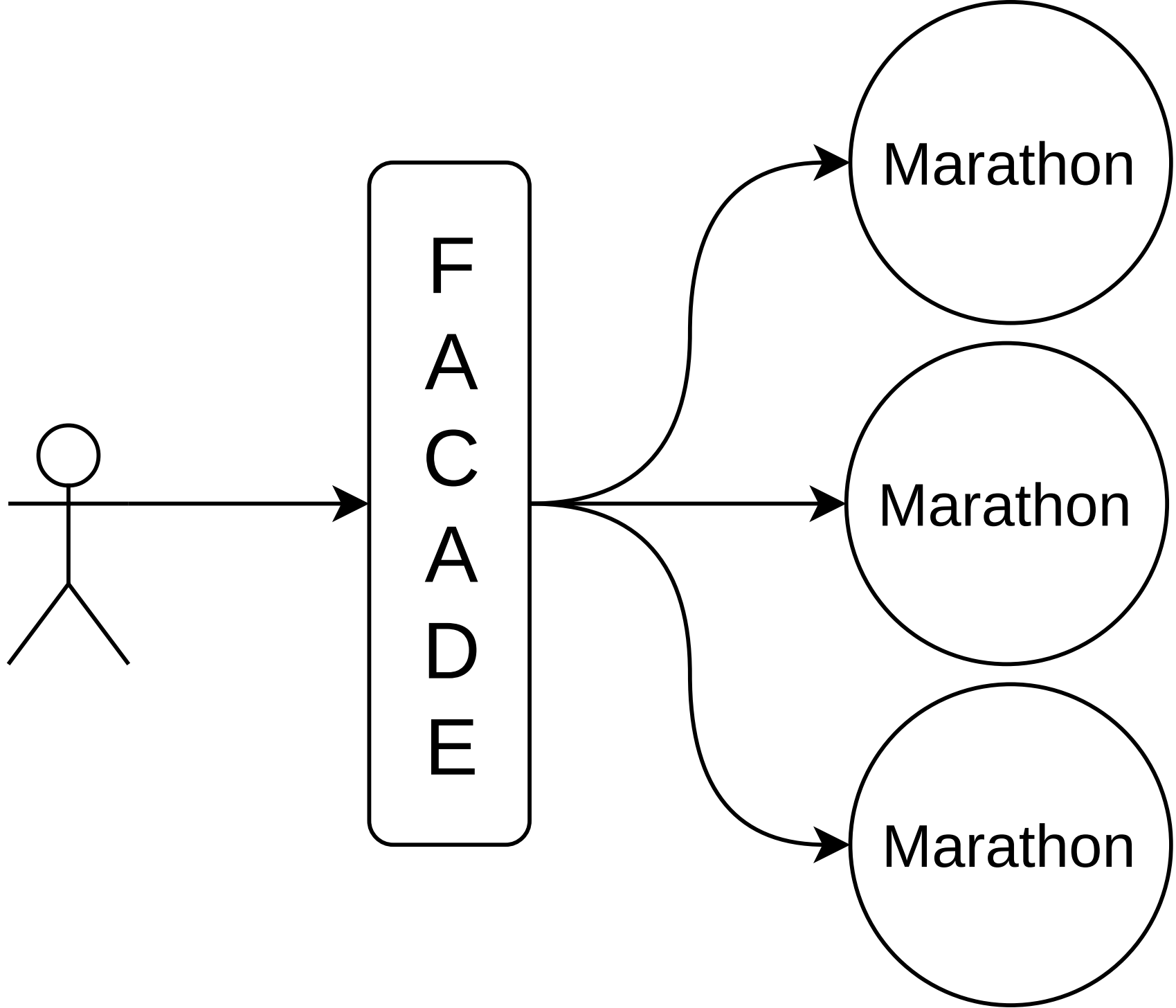
# **#7: PREFER BATCHING**





# **#8: SHARD YOUR MARATHON**







# SUMMARY

1. Monitor — enable metrics
2. Tune JVM
3. Optimize Zookeeper
4. Update to 1.3.13\*
5. Do not use the event bus
6. Use a custom executor
7. Prefer batching
8. Shard your marathon