TAPAS Final Talk

St. Gallen, 19. December 2022

Atilla Güven, Fabian Gubler, Fabio Göldi, Phil Natter, Valentin Berger

ASSE – Group 3

University of St.Gallen

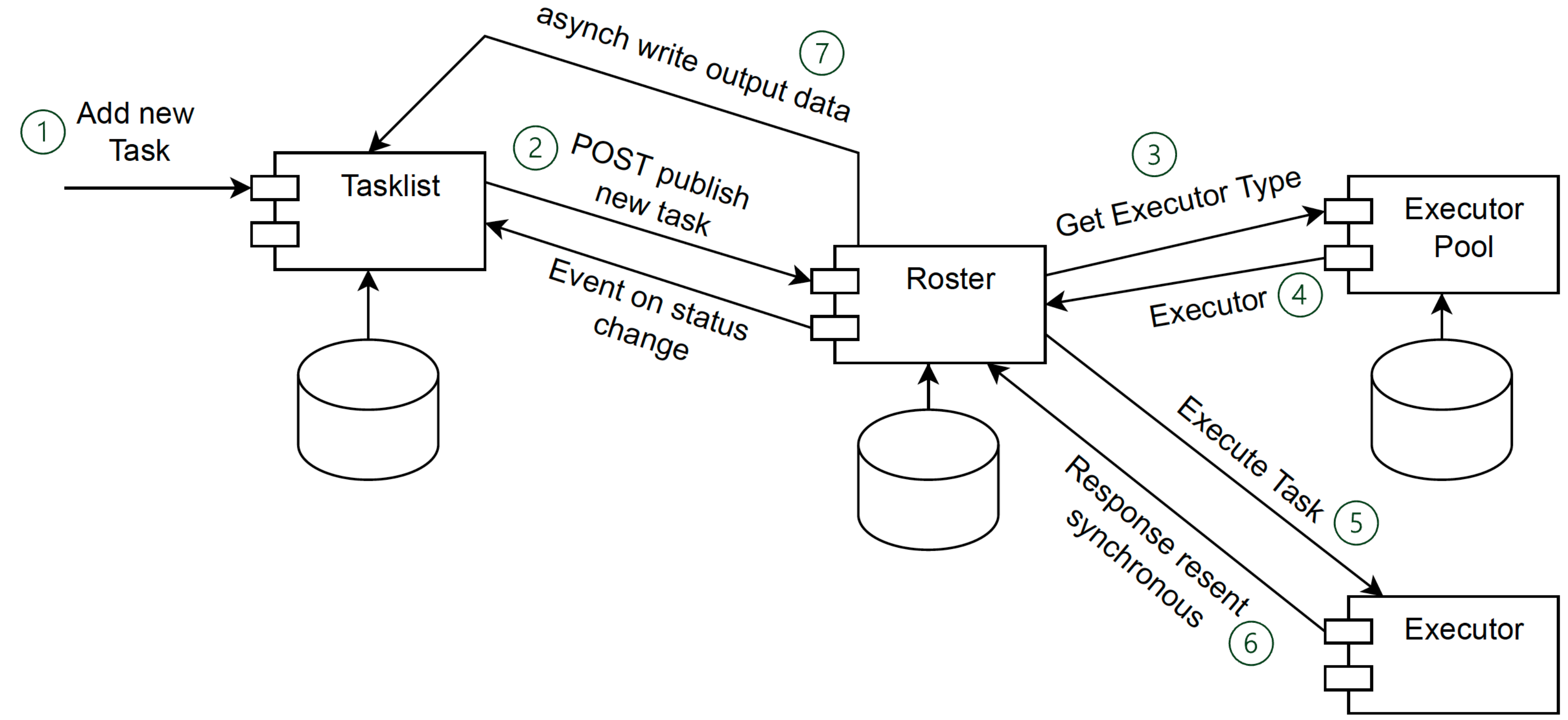
Agenda

1. Overview
2. Architecture
3. Testing
4. Demo
5. Outlook & Reflection

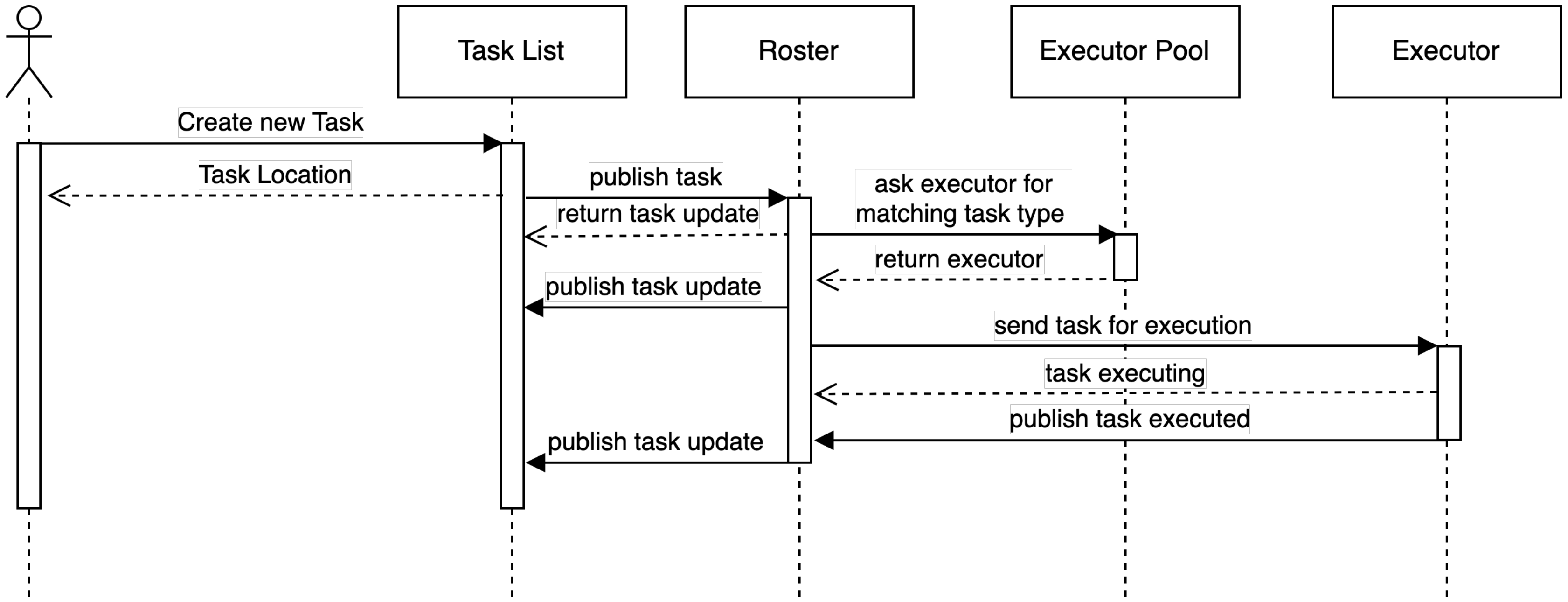
Overview

Sequence Diagram

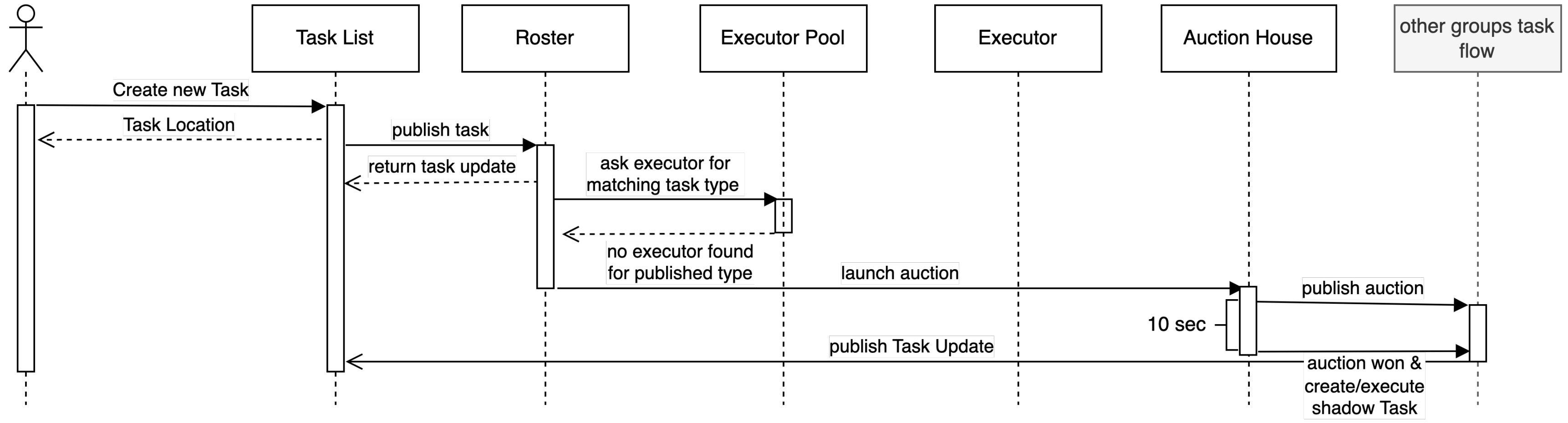
Old Workflow



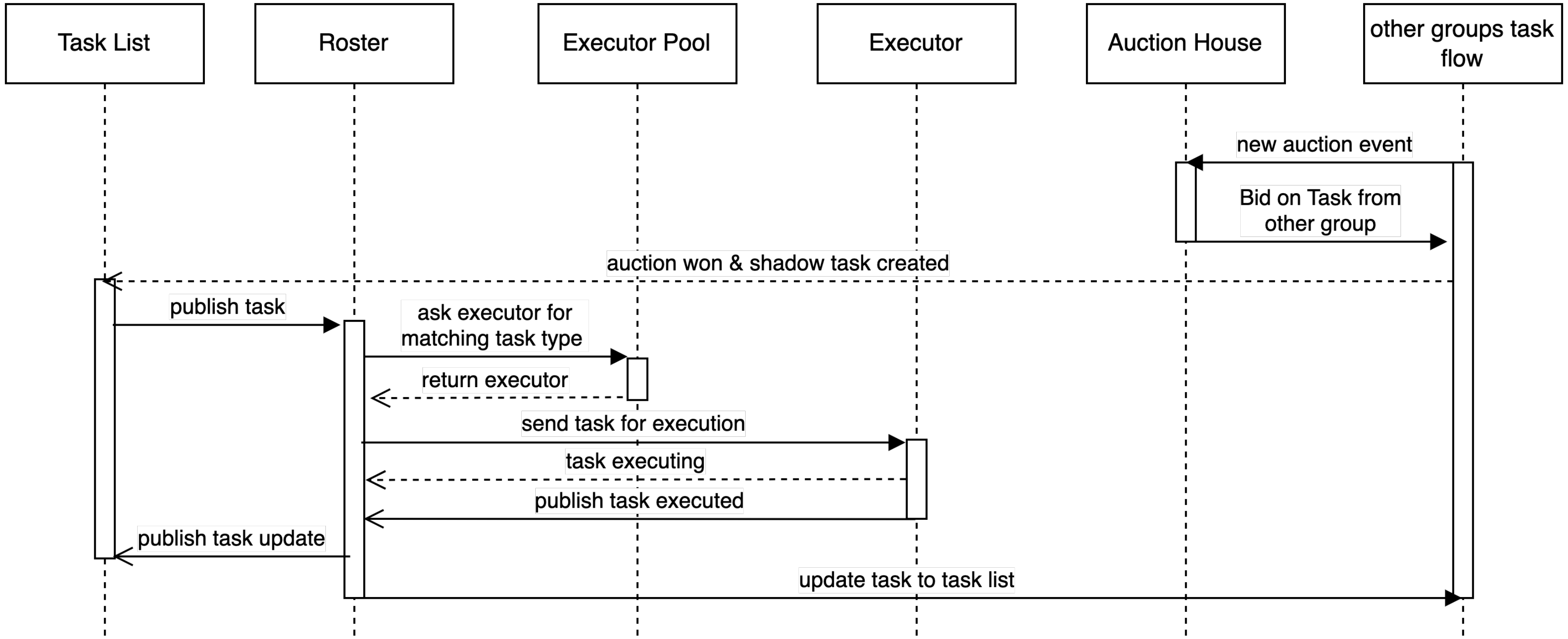
Local Task Execution



Task Execution via Auction House



Task Execution for Other Groups



Architecture

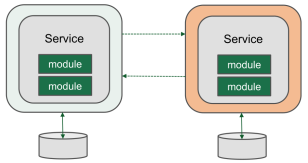
ADRs & Decoupling

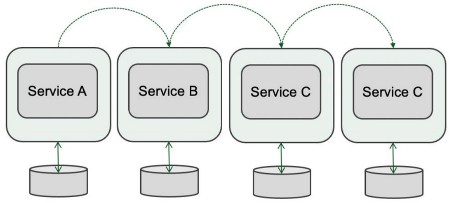
«Why is more important than how. »

- M. Richards and N. Ford, ”Fundamentals of Software Architecture - An Engineering Approach.”

Architectural Decisions

Decisions taken since break





Asynchronous Communication

- + Strenghtens our performance
- + Improve Scalability
- - Increase complexity

Choreographic workflow

- + Reduce decoupling of services
- + Improve Fault tolerance
- - Complicates error handling & state management

Auction House

+ Handle incoming Tasks

+ Handle incoming Auctions

Decoupling Components

Steps taken in our implementation

Hypermedia-based Discovery

- **Crawler algorithm**, implemented to discover our "business clients"
- **URLs are no longer "hardcoded"** in a central registry, but found during run-time
- Services **decoupled** from other groups

Fairly easy for new "business clients" to connect

Additionally, we have gained important property "scalability"

Semantic Hypermedia

Hardcoded: **Base URI** to hypermedia search engine & SPARQL query sent to endpoint

Semantic: Retrieval of **configuration** and **action affordances** during run-time

- Services **decoupled** from devices

Client does not need to know all API Endpoints

Able to change behavior or state depending on Resources

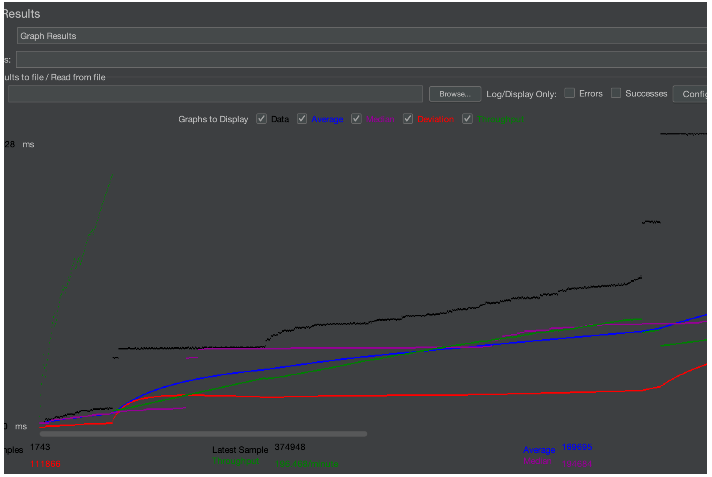
SOURCE: (2018, Reddy and Minnaar), (2019, Clautice)

Testing

JMeter Load Testing

Testing VM – create 5000 tasks – Stress Test

before



Results

Graph Results

ults to file / Read from file

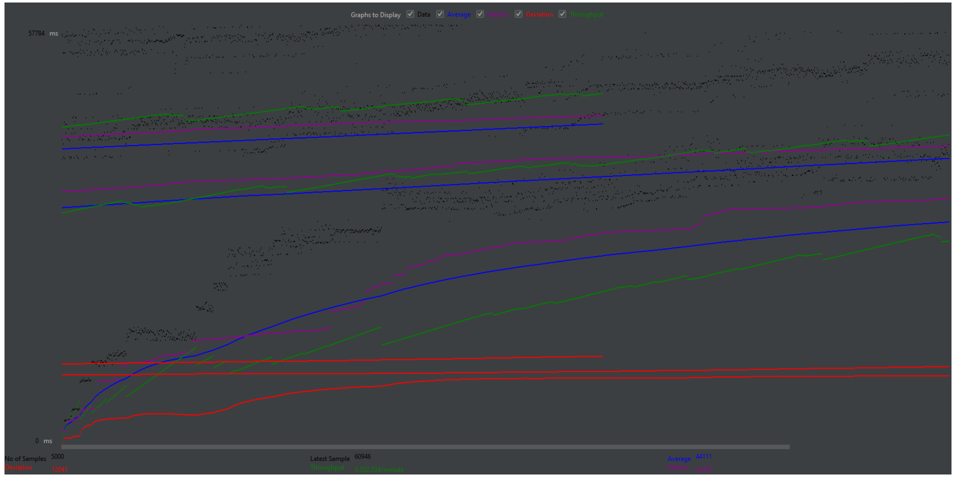
Browse... Log/Display Only: ☐ Errors ☐ Successes Config

Graphs to Display ✓ Data ✓ Average ✓ Median ✓ Deviation ✓ Throughput

```
nples 1743          Latest Sample 374948          Average 169695
      111866        Throughput 196.468/minute     Median  194684
```

- high latency
- stopped at thread 1743
- VM crashed

after



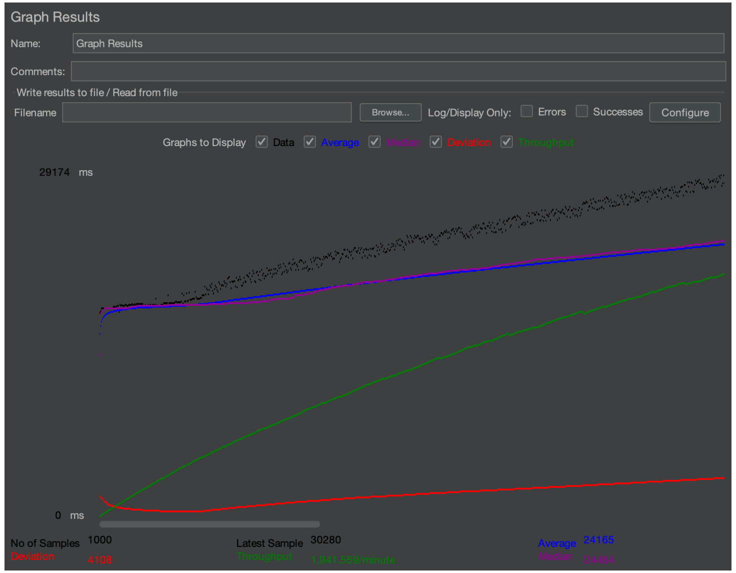
Graphs to Display ✓ Data ✓ Average ✓ Median ✓ Deviation ✓ Throughput

```
No of Samples 5000     Latest Sample 60946             Average 44111
Deviation     12041    Throughput    3,332.334/minute  Median  43239
```

- Throughput 55 requests/sec
- 0 Errors

Testing VM – add 1000 executors – Stress Test

before


Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☐ Successes Configure

Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

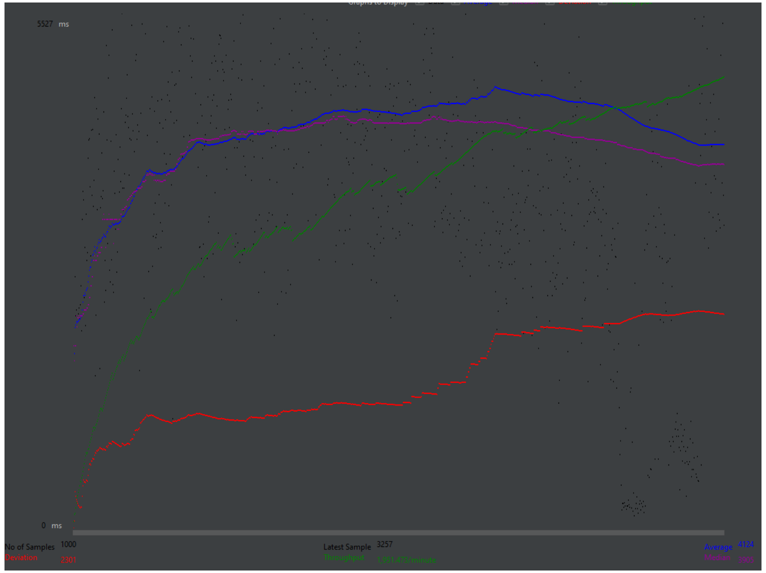
29174 ms

0 ms

No of Samples 1000 | Latest Sample 30280 | Average 24165

Deviation 4108 | Throughput 1,941.559/minute | Median 24464

after


5527 ms

0 ms

No of Samples 1000 | Latest Sample 3257 | Average 4124

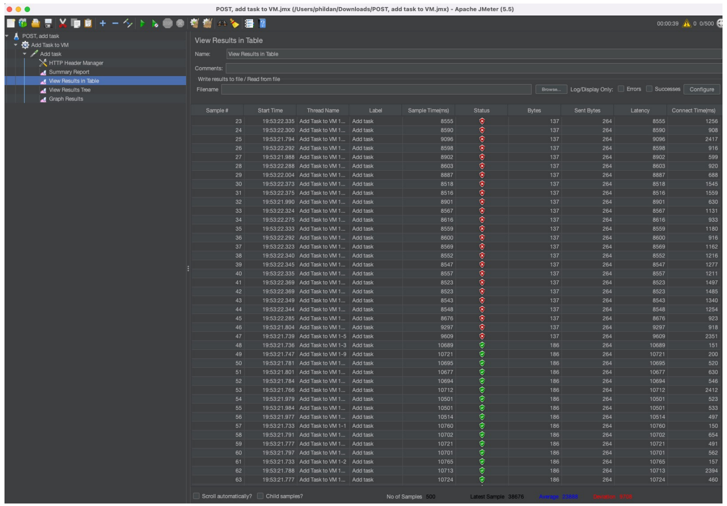
Deviation 2301 | Throughput 1,951.473/minute | Median 3905

3x higher throughput – 31.6 requests/sec
6x better latency
0 Request errors

Better workflow

- the first requests ended in an error
- same problem with add executor
- Lot of requests returned 500 errors on tests with more than 1000 requests
- Errors are all fixed

before



POST, add task to VM.jmx (/Users/phildan/Downloads/POST, add task to VM.jmx) - Apache JMeter (5.5)

00:00:39 0 0/500

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

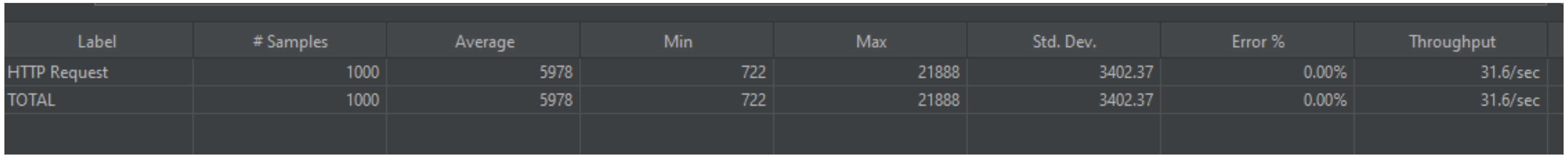
Filename

Log/Display Only: Errors Successes

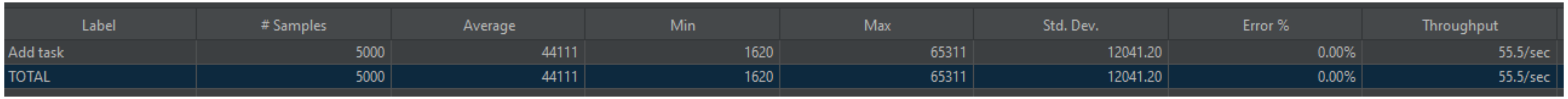
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
23	19:53:22.335	Add Task to VM 1...	Add task	8555	✗	137	264	8555	1256
24	19:53:22.300	Add Task to VM 1...	Add task	8590	✗	137	264	8590	908
25	19:53:21.794	Add Task to VM 1...	Add task	9096	✗	137	264	9096	2417
26	19:53:22.292	Add Task to VM 1...	Add task	8598	✗	137	264	8598	916
27	19:53:21.988	Add Task to VM 1...	Add task	8902	✗	137	264	8902	599
28	19:53:22.288	Add Task to VM 1...	Add task	8603	✗	137	264	8603	920
29	19:53:22.004	Add Task to VM 1...	Add task	8887	✗	137	264	8887	688
30	19:53:22.373	Add Task to VM 1...	Add task	8518	✗	137	264	8518	1545
31	19:53:22.375	Add Task to VM 1...	Add task	8516	✗	137	264	8516	1559
32	19:53:21.990	Add Task to VM 1...	Add task	8901	✗	137	264	8901	630
33	19:53:22.324	Add Task to VM 1...	Add task	8567	✗	137	264	8567	1131
34	19:53:22.275	Add Task to VM 1...	Add task	8616	✗	137	264	8616	933
35	19:53:22.333	Add Task to VM 1...	Add task	8559	✗	137	264	8559	1180
36	19:53:22.292	Add Task to VM 1...	Add task	8600	✗	137	264	8600	916
37	19:53:22.323	Add Task to VM 1...	Add task	8569	✗	137	264	8569	1162
38	19:53:22.340	Add Task to VM 1...	Add task	8552	✗	137	264	8552	1216
39	19:53:22.345	Add Task to VM 1...	Add task	8547	✗	137	264	8547	1277
40	19:53:22.335	Add Task to VM 1...	Add task	8557	✗	137	264	8557	1211
41	19:53:22.369	Add Task to VM 1...	Add task	8523	✗	137	264	8523	1497
42	19:53:22.369	Add Task to VM 1...	Add task	8523	✗	137	264	8523	1485
43	19:53:22.349	Add Task to VM 1...	Add task	8543	✗	137	264	8543	1340
44	19:53:22.344	Add Task to VM 1...	Add task	8548	✗	137	264	8548	1254
45	19:53:22.285	Add Task to VM 1...	Add task	8676	✗	137	264	8676	923
46	19:53:21.804	Add Task to VM 1...	Add task	9297	✗	137	264	9297	918
47	19:53:21.739	Add Task to VM 1-5	Add task	9609	✗	137	264	9609	2351
48	19:53:21.736	Add Task to VM 1-3	Add task	10689	✓	186	264	10689	151
49	19:53:21.747	Add Task to VM 1-9	Add task	10721	✓	186	264	10721	200
50	19:53:21.781	Add Task to VM 1...	Add task	10695	✓	186	264	10695	520
51	19:53:21.801	Add Task to VM 1...	Add task	10677	✓	186	264	10677	630
52	19:53:21.784	Add Task to VM 1...	Add task	10694	✓	186	264	10694	546
53	19:53:21.766	Add Task to VM 1...	Add task	10712	✓	186	264	10712	2412
54	19:53:21.979	Add Task to VM 1...	Add task	10501	✓	186	264	10501	523
55	19:53:21.984	Add Task to VM 1...	Add task	10501	✓	186	264	10501	533
56	19:53:21.977	Add Task to VM 1...	Add task	10514	✓	186	264	10514	497
57	19:53:21.733	Add Task to VM 1-1	Add task	10760	✓	186	264	10760	150
58	19:53:21.791	Add Task to VM 1...	Add task	10702	✓	186	264	10702	654
59	19:53:21.777	Add Task to VM 1...	Add task	10721	✓	186	264	10721	491
60	19:53:21.797	Add Task to VM 1...	Add task	10701	✓	186	264	10701	562
61	19:53:21.733	Add Task to VM 1-2	Add task	10765	✓	186	264	10765	157
62	19:53:21.788	Add Task to VM 1...	Add task	10713	✓	186	264	10713	2394
63	19:53:21.777	Add Task to VM 1...	Add task	10724	✓	186	264	10724	460

Scroll automatically? Child samples? No of Samples 500 Latest Sample 38676 Average 23888 Deviation 9708

after



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
HTTP Request	1000	5978	722	21888	3402.37	0.00%	31.6/sec
TOTAL	1000	5978	722	21888	3402.37	0.00%	31.6/sec



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
Add task	5000	44111	1620	65311	12041.20	0.00%	55.5/sec
TOTAL	5000	44111	1620	65311	12041.20	0.00%	55.5/sec

Demo

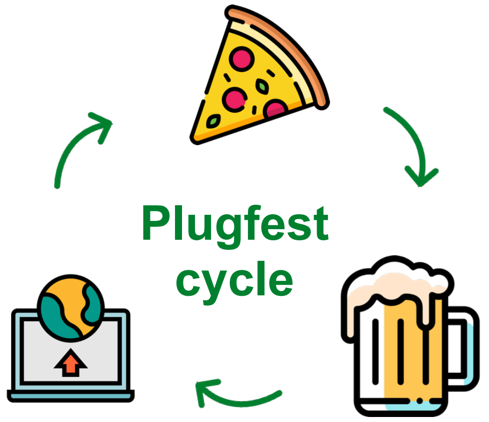
Creating an Executor & Receive Tasks

Outlook - improvements

- General improvement of stability and error handling (fault tolerance)
- Scheduling of tasks to rerun failed tasks (workflow)
  - Assigning of tasks to executors (no bid or executor)
  - Removing of unassigned/failed tasks
- Assigning of tasks with multiple executors of same type (workflow)
  - Prequisity would be to track executions

Group Project Experience

- Diverse backgrounds presented challenges and benefits
- We split tasks based on individual strengths
- Pair programming helped a lot and allowed to share knowledge and improve skills
- Highlight was the plugfest – Pizza, beer and redeployment
- Steep learning curve for everyone



Questions?
Feedback?