PyCon China 2020

PyConChina 2020
PYTHON 中国开发者大会 2020

Python For Good

智能问答系统在西山居中的落地

黄鸿波

西山居游戏

自我介绍

- 2008年获得高级职业资格证
- 2012年开始接触AI，以Numenta公司的HTM（Hierarchical Temporal Memory）算法入门AI
- 2016年开始研究深度学习框架TensorFlow
- 2018年出版《TensorFlow进阶指南 基础、算法与应用》一书
- 2020年获得谷歌开发者专家称号（GDE,Google Developer Expert）
- 目前在西山居技术中心负责AI团队建设

Content

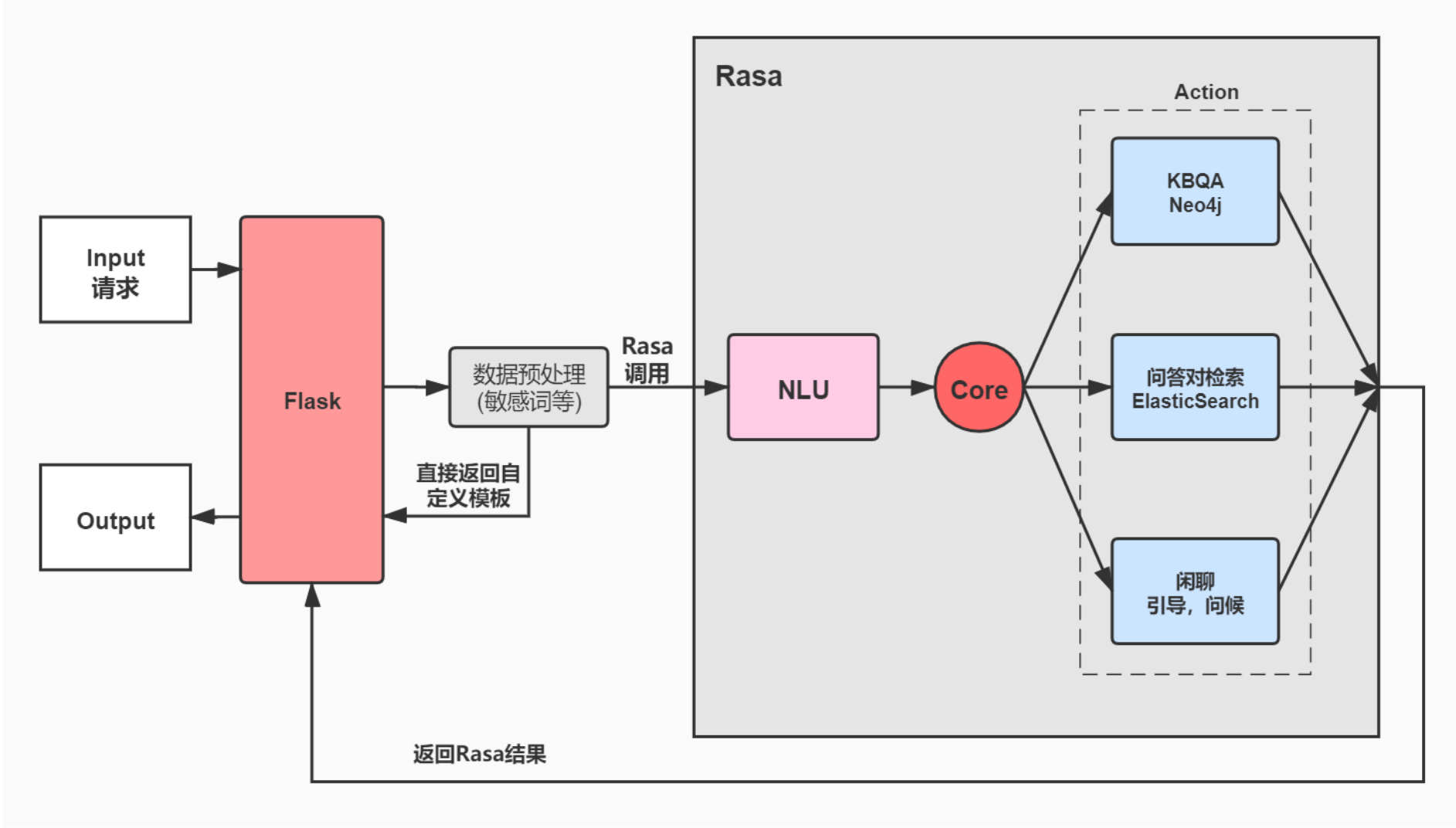
传统问答与智能问答

传统问答与智能问答

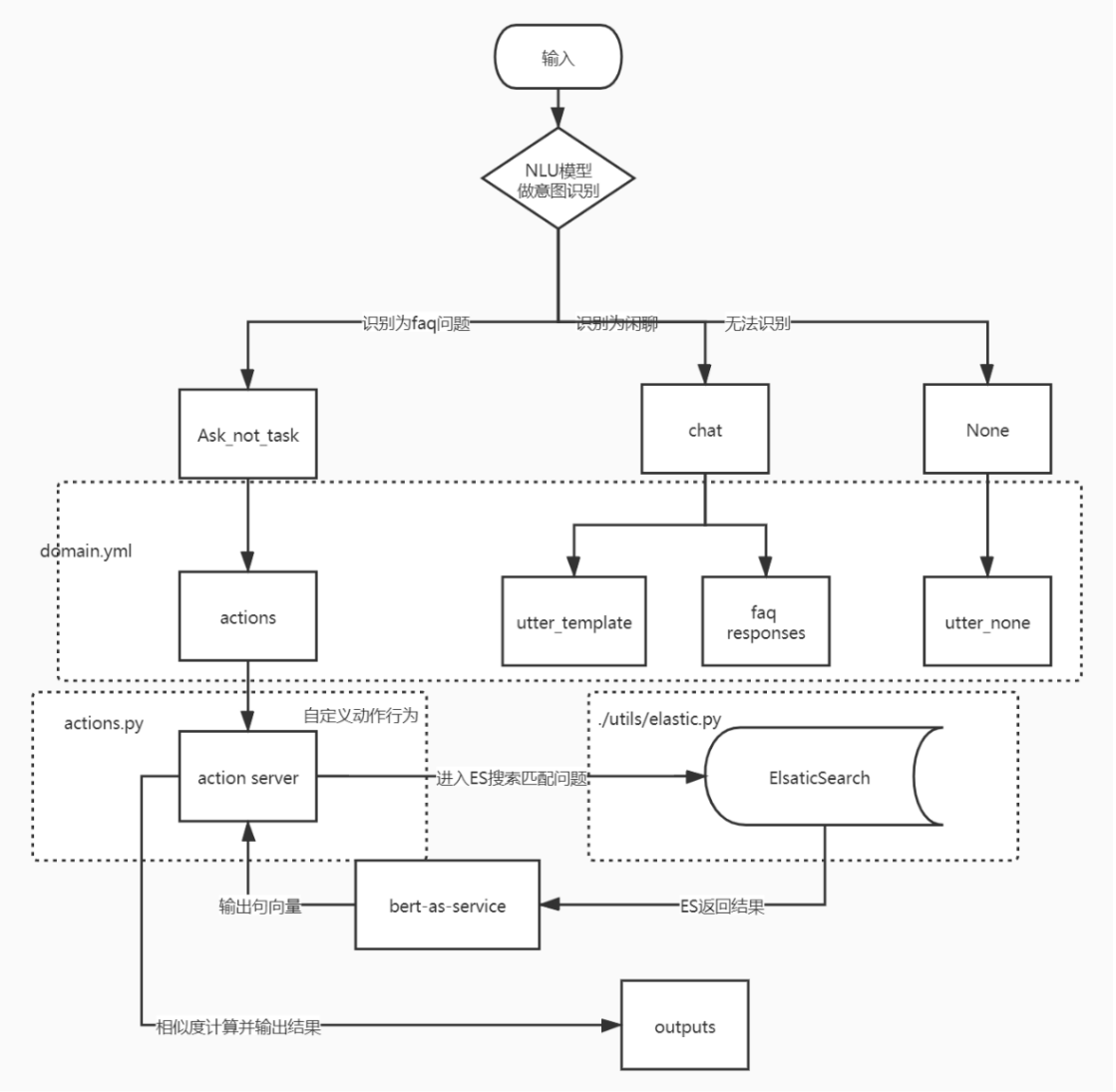
	传统问答	智能问答
灵活性	死板	灵活
实现方式	关键词检索	NLP与检索混合
对错词的容错性	差	强
语义理解	无	有

西山居智能问答部署方案

西山居智能问答部署方案



西山居智能问答部署方案

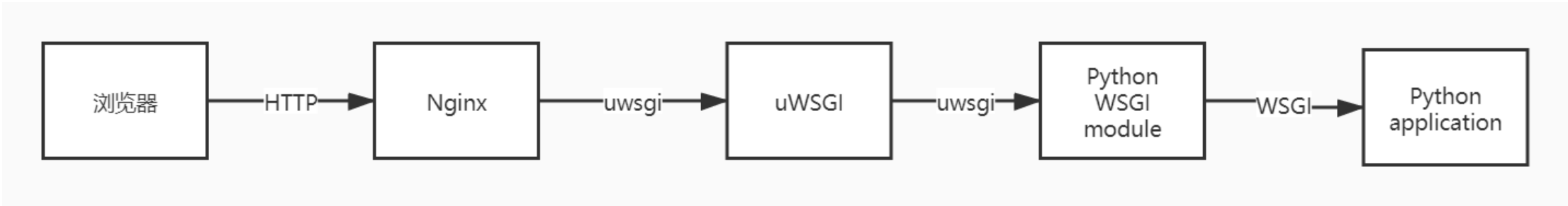


西山居智能问答部署方案——Flask

uWSGI

Flask

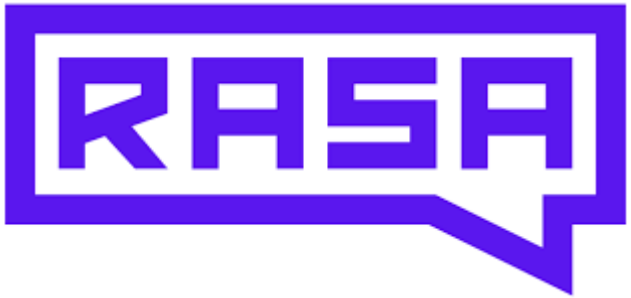
西山居智能问答部署方案——Flask



Python For Good
PyCon China 2020 | PYTHON 中国开发者大会 2020

智能问答与Python多线程

智能问答与Python多线程



+ sanic

智能问答与Python多线程

server.py

root > miniconda3 > envs > deep > lib > python3.7 > site-packages > sanic > server.py > serve

```
970         """
971         server_settings["reuse_port"] = True
972         server_settings["run_multiple"] = True
973
974         # Handling when custom socket is not provided.
975         if server_settings.get("sock") is None:
976             sock = socket()
977             sock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
978             sock.bind((server_settings["host"], server_settings["port"]))
979             sock.set_inheritable(True)
980             server_settings["sock"] = sock
981             server_settings["host"] = None
982             server_settings["port"] = None
983
984         processes = []
985
986         def sig_handler(signal, frame):
987             logger.info("Received signal %s. Shutting down.", Signals(signal).name)
988             for process in processes:
989                 os.kill(process.pid, SIGTERM)
990
991         signal_func(SIGINT, lambda s, f: sig_handler(s, f))
992         signal_func(SIGTERM, lambda s, f: sig_handler(s, f))
993
994         for _ in range(workers):
995             process = Process(target=serve, kwargs=server_settings)
996             process.daemon = True
997             process.start()
998             processes.append(process)
999
1000        for process in processes:
1001            process.join()
1002
1003        # the above processes will block this until they're stopped
1004        for process in processes:
1005            process.terminate()
1006        server_settings.get("sock").close()
1007
```

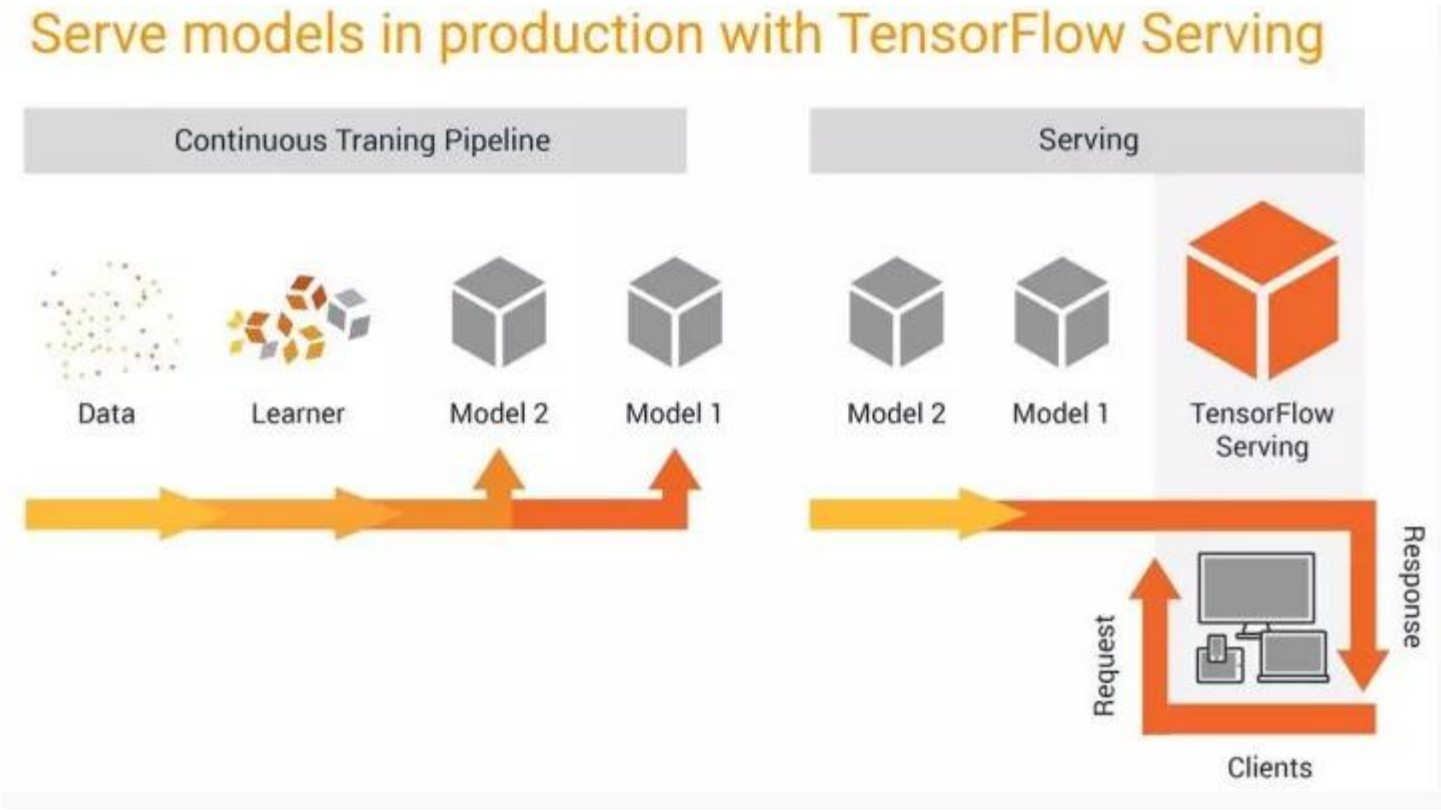
sanic与pytorch冲突

智能问答与Python多线程

bert-as-service or TensorFlow Serving

解决大规模QPS问题

解决大规模QPS问题


Serve models in production with TensorFlow Serving
Continuous Traning Pipeline
Serving
Data
Learner
Model 2
Model 1
Model 2
Model 1
TensorFlow Serving
Request
Response
Clients

解决大规模QPS问题


```
max_batch_size { value: 128 }
batch_timeout_micros { value: 0 }
max_enqueued_batches { value: 1000000 }
num_batch_threads { value: 8 }
```

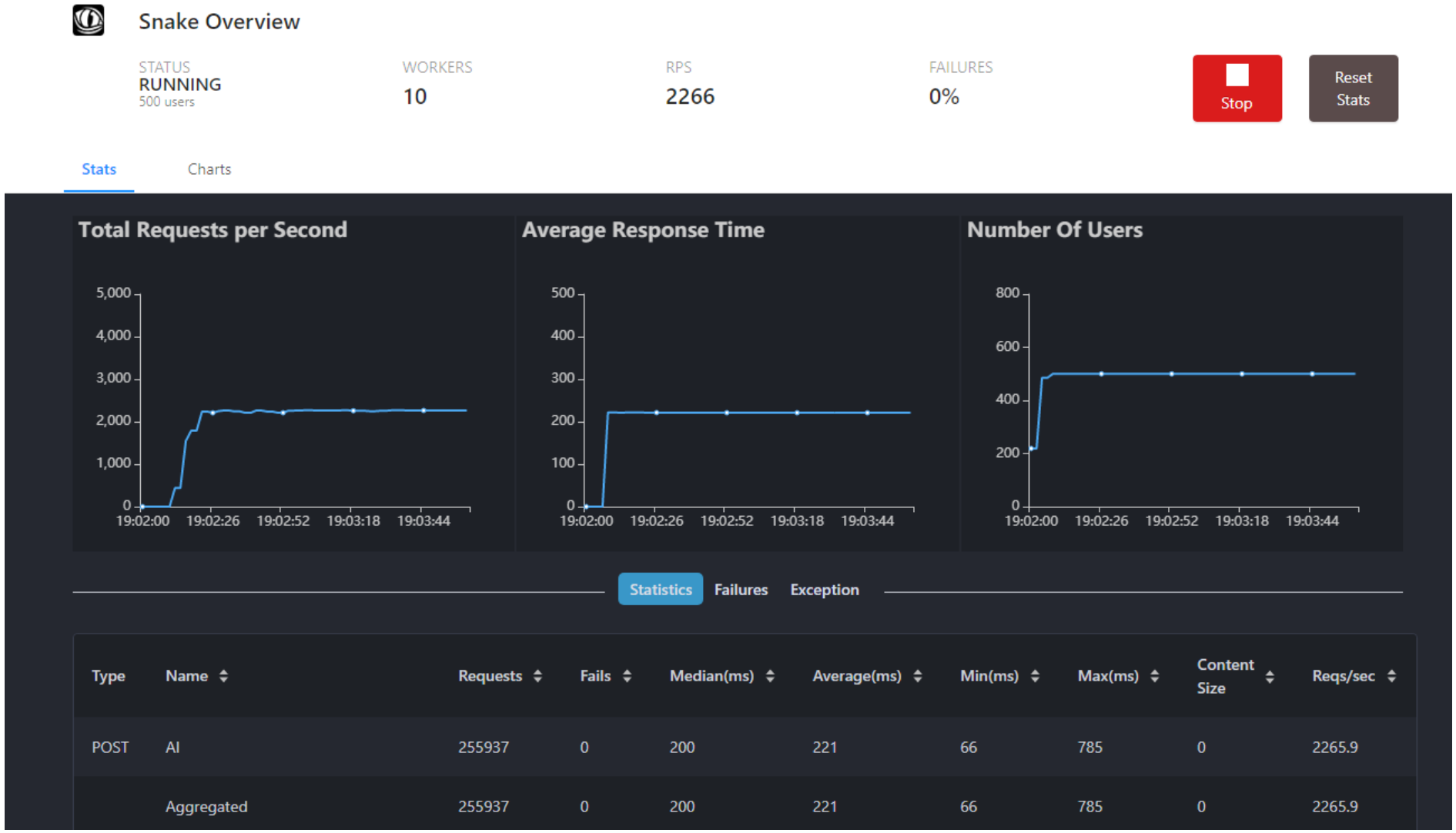
max_batch_size：任何批次的最大大小。此参数控制吞吐量/等待时间的折衷，还避免了批处理过大以至于超过了某些资源限制（例如，GPU内存来保存批处理的数据）。

batch_timeout_micros：执行批处理之前需要等待的最长时间（即使尚未达到max_batch_size）。用于控制尾部延迟。（有关basic_batch_scheduler.h确切的延迟合同，请参阅。）

max_enqueued_batches：可以排入调度程序的批处理任务的数量。通过拒绝需要很长时间才能到达的请求，而不是建立大量积压，来限制排队延迟。

num_batch_threads：并行度，即并发处理的最大批次数。

解决大规模QPS问题


Snake Overview
STATUS
RUNNING
500 users
WORKERS
10
RPS
2266
FAILURES
0%
Stop
Reset Stats
Stats
Charts
Total Requests per Second
5,000
4,000
3,000
2,000
1,000
0
19:02:00
19:02:26
19:02:52
19:03:18
19:03:44
Average Response Time
500
400
300
200
100
0
19:02:00
19:02:26
19:02:52
19:03:18
19:03:44
Number Of Users
800
600
400
200
0
19:02:00
19:02:26
19:02:52
19:03:18
19:03:44
Statistics
Failures
Exception
Type
Name
Requests
Fails
Median(ms)
Average(ms)
Min(ms)
Max(ms)
Content Size
Reqs/sec
POST
AI
255937
0
200
221
66
785
0
2265.9
Aggregated
255937
0
200
221
66
785
0
2265.9

PyCon China 2020

Python For Good

PyConChina 2020 | PYTHON 中国开发者大会 2020

THANK YOU