



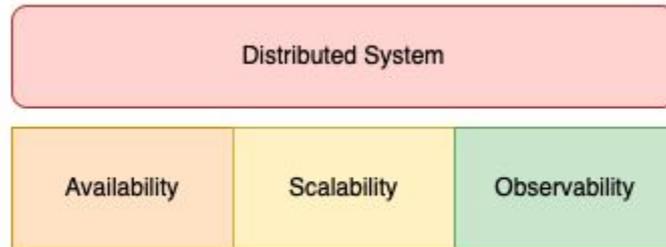
Introduction To Observability

April 15, 2022

Henry Wu

Why Observability?

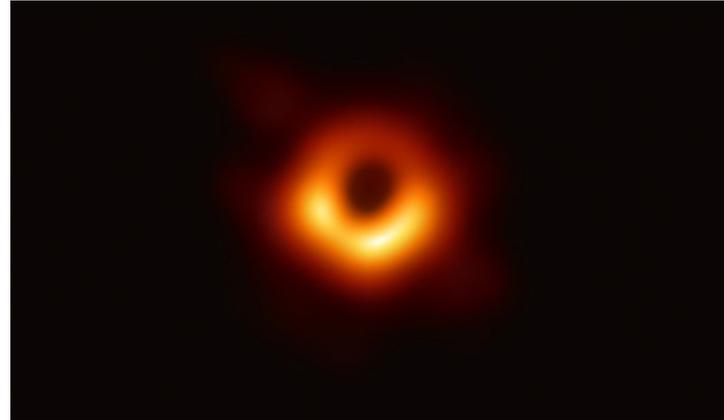
Three Pillars of Large-Scale Distributed System



What is Observability?

Apr 10, 2019 Black Hole Image Makes History;
NASA Telescopes Coordinated Observations

https://www.nasa.gov/mission_pages/chandra/news/black-hole-image-makes-history



Monitoring vs. Observability



Good monitoring is a staple of high-performing teams. [DevOps Research and Assessment \(DORA\) research](#) shows that a comprehensive monitoring and observability solution, along with a number of other technical practices, positively contributes to [continuous delivery](#).

DORA's research defined these terms as follows:

Monitoring is tooling or a technical solution that allows teams to watch and understand the state of their systems. Monitoring is based on gathering predefined sets of metrics or logs.

Observability is tooling or a technical solution that allows teams to actively debug their system. Observability is based on exploring properties and patterns **not defined in advance**.

<https://cloud.google.com/architecture/devops/devops-measurement-monitoring-and-observability>

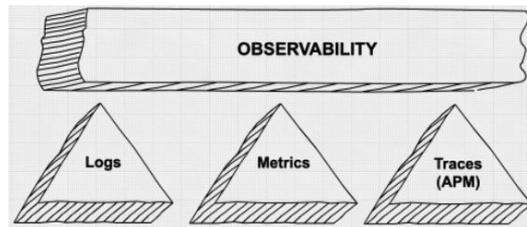
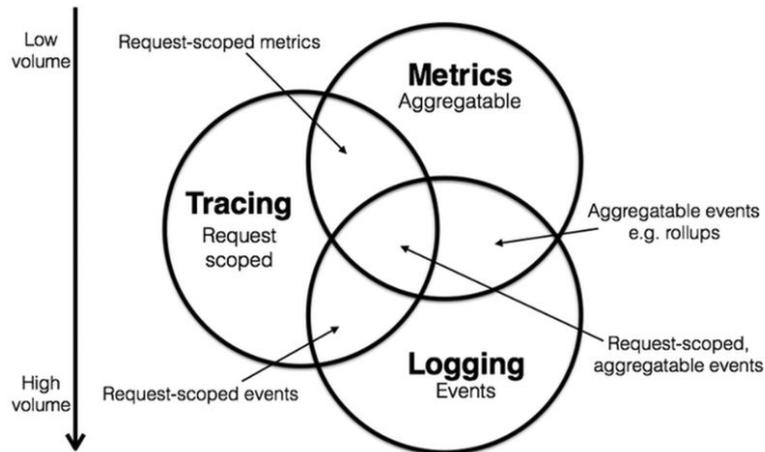
What is Observability?

Observability is the ability to measure the internal states of a system by examining its outputs. A system is considered “observable” if the current state can be estimated by only using information from outputs, namely sensor data.

Observability uses three types of telemetry data — **logs, metrics, and traces** — to provide deep visibility into distributed systems and allow teams to get to the root cause of a multitude of issues and improve the system’s performance.

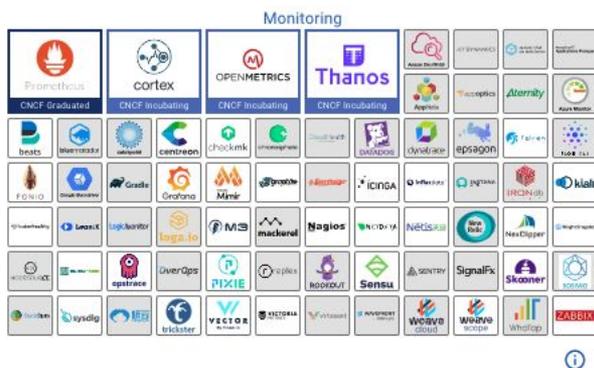
Data formats:

- Metrics (monitoring)
- Span (tracing)
- Free-form (logging)



What is Observability?

CNCF Projects



Opensource

- Monitoring: prometheus, Grafana
- Logging: fluentd, ELK(Elasticsearch, Logstash, Kibana)
- Tracing: jaeger, zipkin

Commercial

- Zabbix
- Lightstep
- Datadog
- APM Services

Offline vs. Online

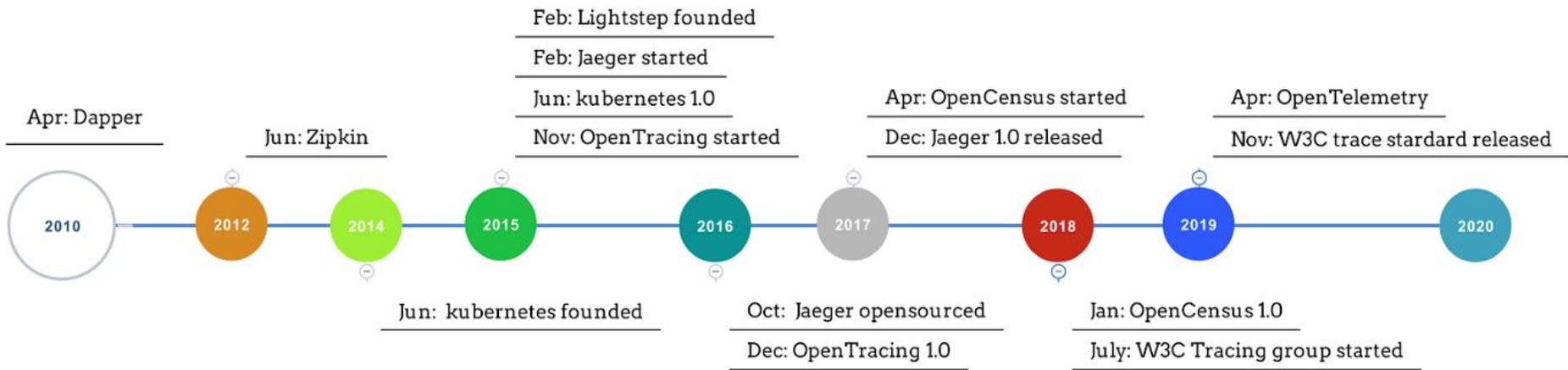


Offline: the traditional way, eg, core dump, profiling, activity monitor...

Online:

- Monolithic
- Distributed
 - Dapper 2010 (Google)
 - Zipkin 2012 (Twitter)
 - Jaeger 2016 (Uber)
 - Skywalking 2018 ...

History - Distributed Tracing



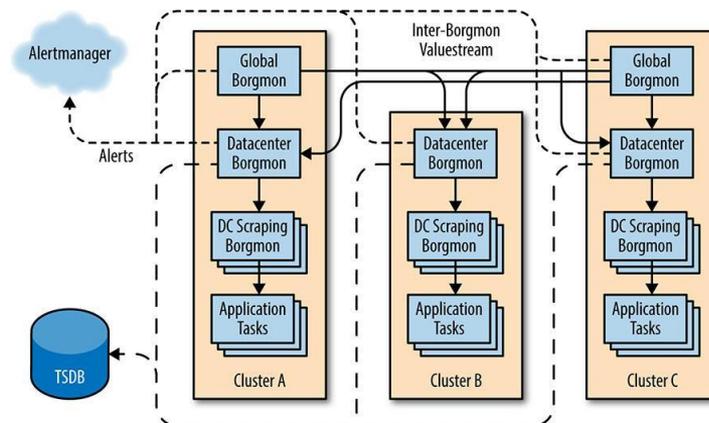
History - Monitoring System

<https://www.sumologic.com/blog/monitoring-tools-history/>

1990s - [Nigel's Monitor](#)(nmon), [Big Brother](#)(BB), MRTG, RRDTOol

2000s - Cacti, Nagios, Zabbix...

2015+ - Prometheus (2nd CNCF project)

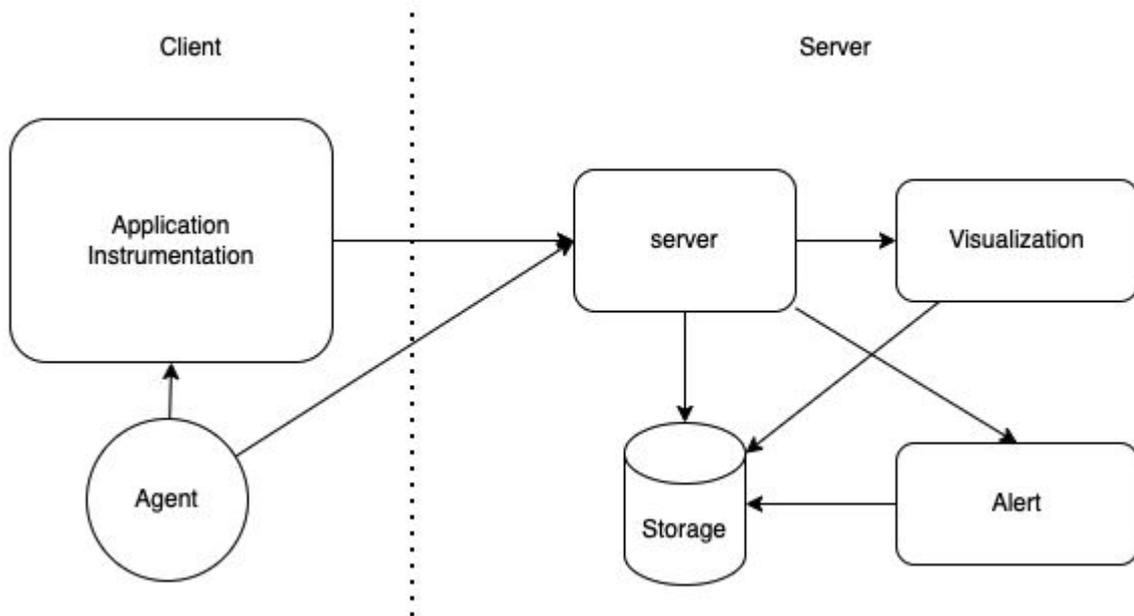




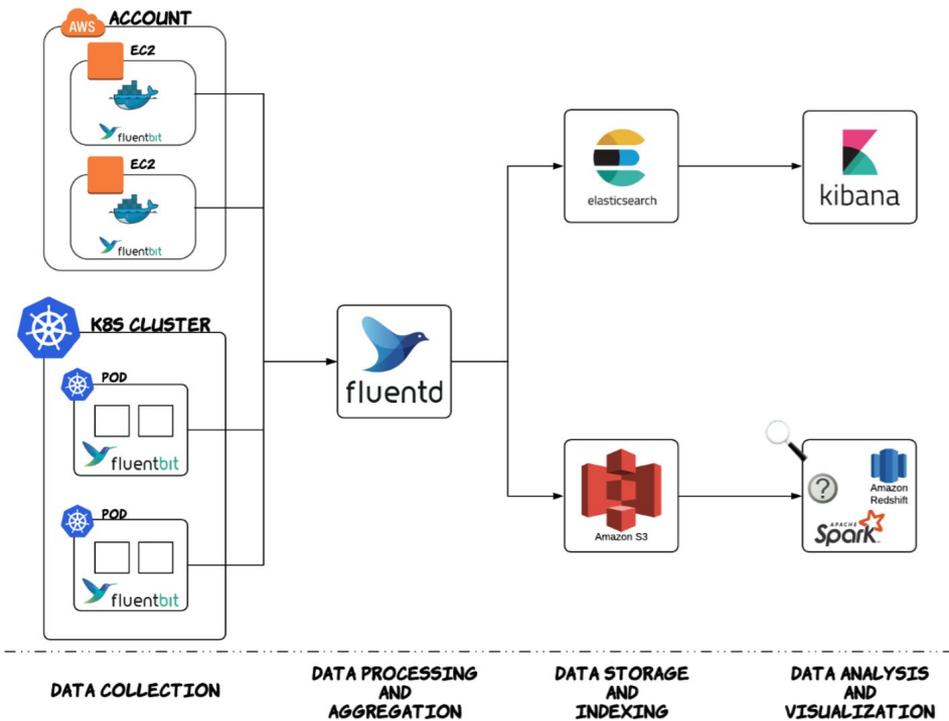
System Architecture



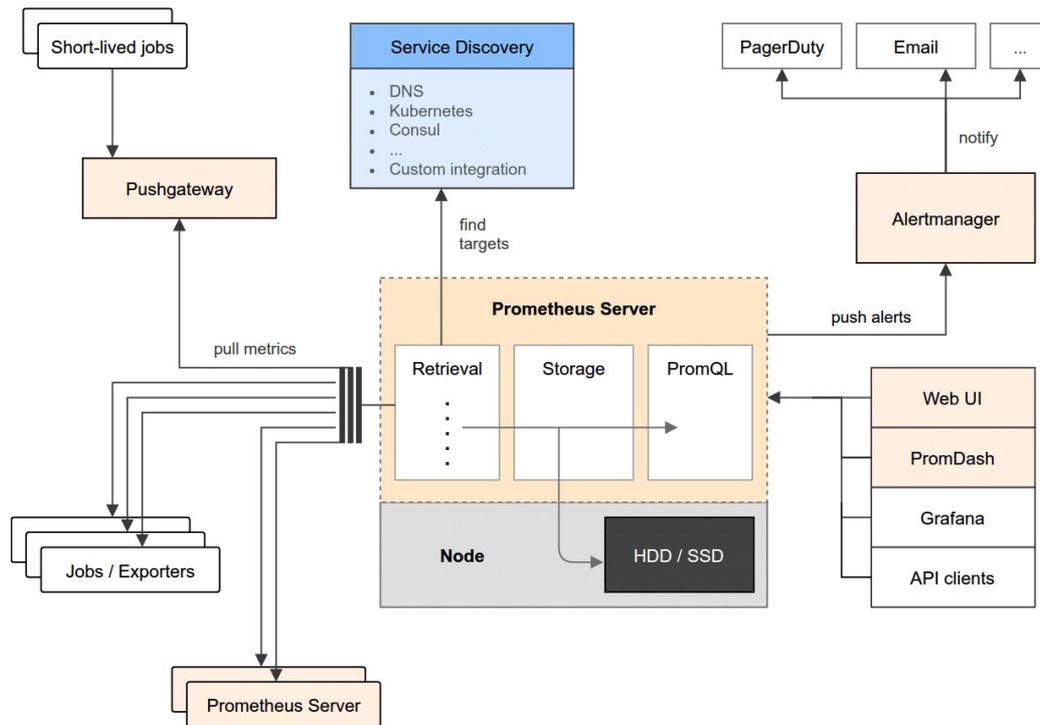
Distributed Online System Architecture



Logging

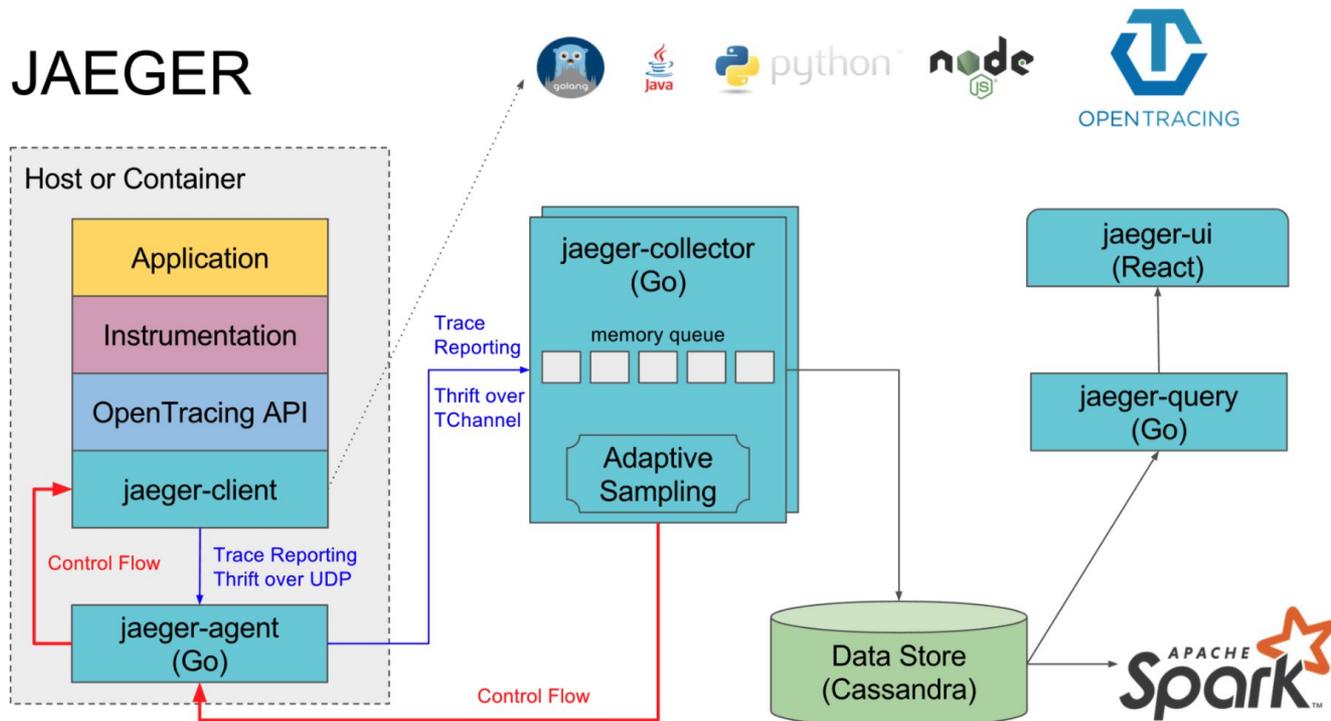


Metrics



Tracing

JAEGER





HiQ - A Modern Observability System

<https://github.com/oracle-samples/hiq>

Client Side Instrumentation



*J. C. Huang, Ph.D.
Professor Emeritus
Department of Computer Science
University of Houston
Houston, Texas*

Paper: Program Instrumentation and Software Testing 1978

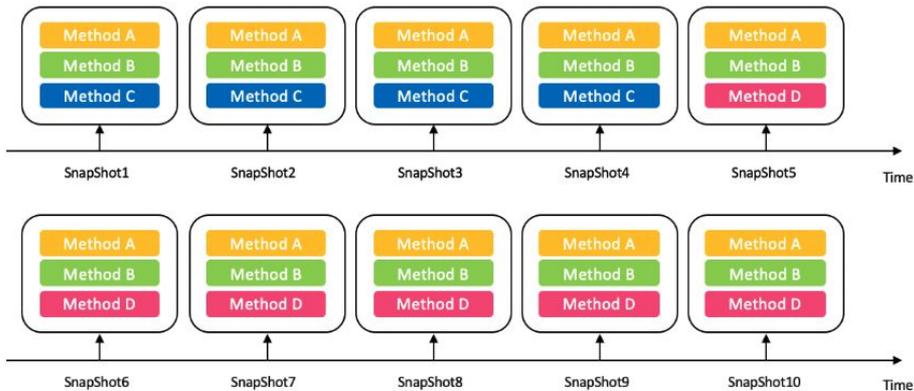
Explicit instrumentation

- openmetrics, micrometer, opentelemetry
- logging(Python), slf4j/log4j (java)
- opentelemetry

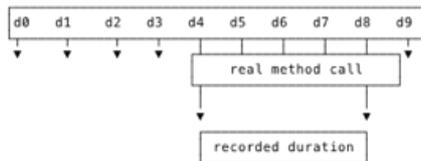
Agent-based instrumentation

- Agent/ exporter
- Thread dump/snapshot

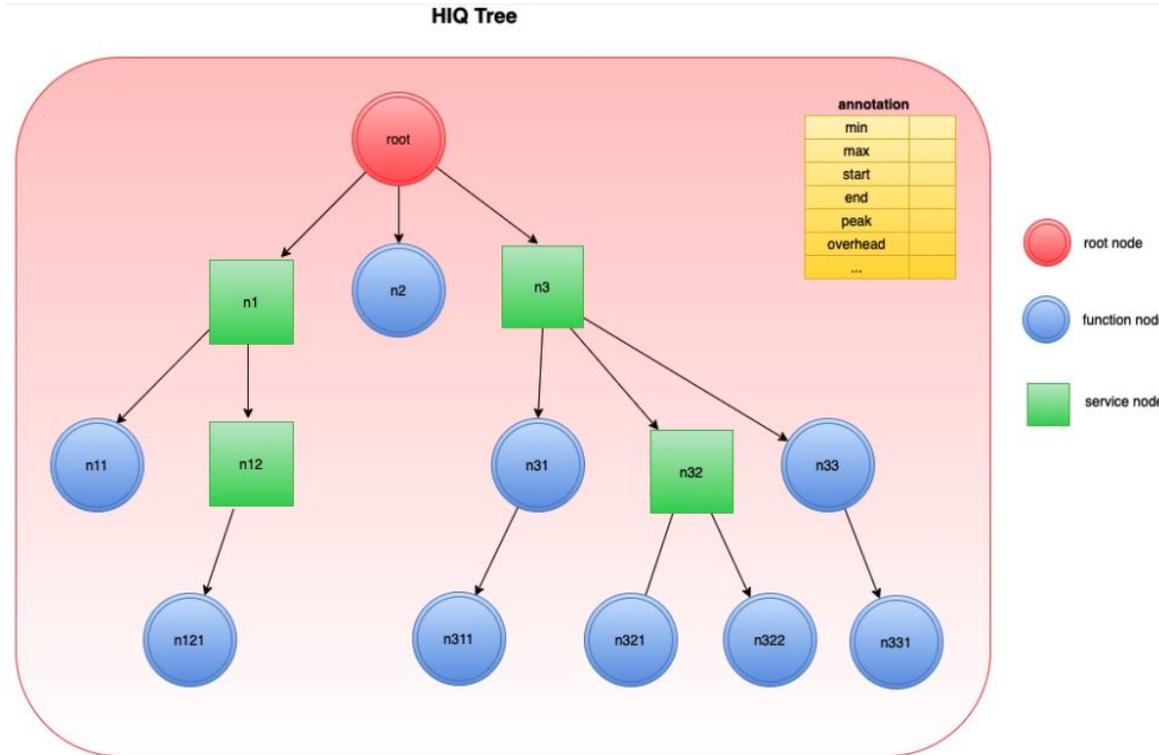
Non-intrusive Instrumentation



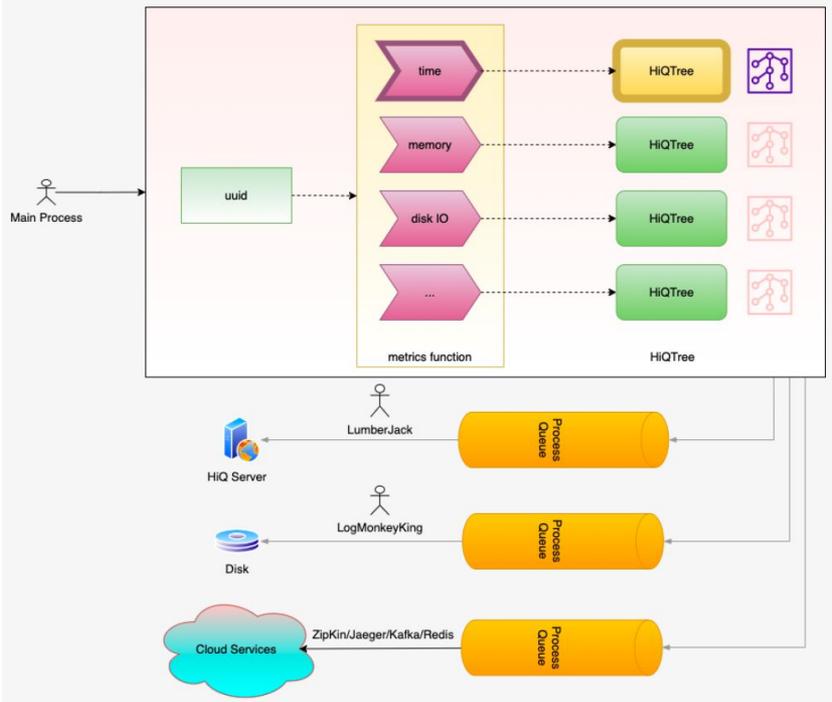
Thread Stack	↔	Duration (ms)	Self Duration (ms)	top slow	Dump Count
Method A		100	0		10
Method B		100	0		10
Method C		40	40		4
Method D		60	60		6



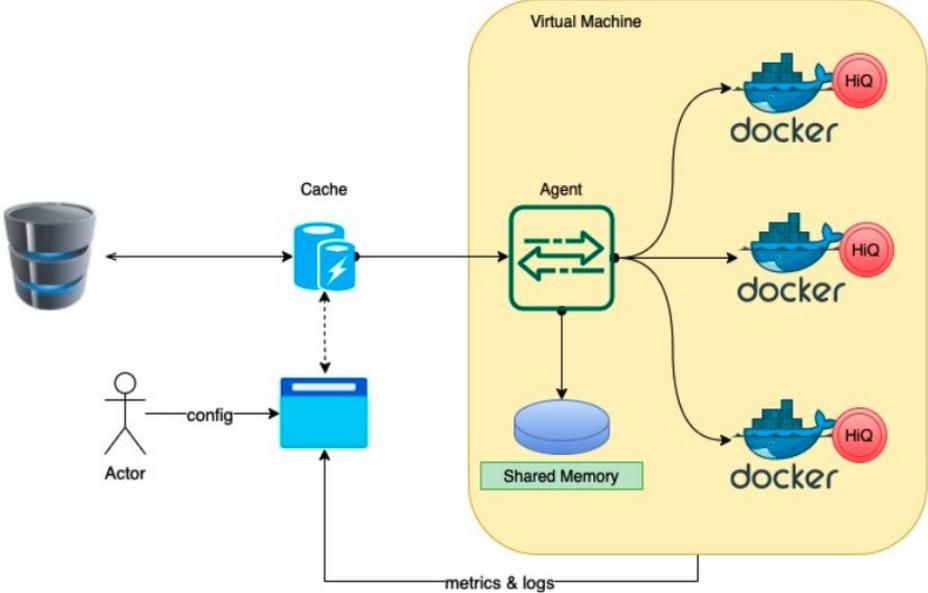
Non-intrusive Instrumentation



HiQ System Architecture for Monolithic Application



HiQ System Architecture for Distributed System



Real Use Case - PaddleOCR

```
1 import os
2 from paddleocr import PaddleOCR, draw_ocr
3 from PIL import Image
4
5 here = os.path.dirname(os.path.realpath(__file__))
6
7 def main():
8     ocr = PaddleOCR(use_angle_cls=True, lang='en')
9     img_path = f'{here}/img_12.jpg'
10    result = ocr.ocr(img_path, cls=True, rec=True)
11    image = Image.open(img_path).convert('RGB')
12    boxes = [line[0] for line in result]
13    txts = [line[1][0] for line in result]
14    scores = [line[1][1] for line in result]
15    im_show = draw_ocr(image, boxes, txts, scores, font_path=f'{here}/simfang.ttf')
16    im_show = Image.fromarray(im_show)
17    im_show.save('result.jpg')
18
19 if __name__ == "__main__":
20     main()
```

```
$time python main.py > /dev/null
```

real	0m2.950s
user	0m8.220s
sys	0m0.628s

The main.py code is from PaddleOCR's official website [chinese-english-model-and-multilingual-model](#)

Real Use Case - PaddleOCR Latency

```
1 import hiq
2 from hiq.framework.paddleocr import PaddleOcrHiQLatency
3
4 driver = PaddleOcrHiQLatency()
5 hiq.mod("main").main()
6 driver.show()
```

```
[2022-04-15 21:40:43.895396 - 21:40:45.273570] [100.00%] ●_root_time(1.3782)
[2022-04-15 21:40:43.895396 - 21:40:44.266277] [ 26.91%] |__init__(0.3709)
[2022-04-15 21:40:43.898308 - 21:40:44.030634] [  9.60%] | |__create_predictor(0.1323)
[2022-04-15 21:40:44.031160 - 21:40:44.150314] [  8.65%] | | |__create_predictor(0.1192)
[2022-04-15 21:40:44.150548 - 21:40:44.266206] [  8.39%] | | |__create_predictor(0.1157)
[2022-04-15 21:40:44.266377 - 21:40:45.273570] [ 73.08%] | |__ocr(1.0072)
[2022-04-15 21:40:44.288023 - 21:40:45.273272] [ 71.49%] | | |__txt_model(0.9852)
[2022-04-15 21:40:44.289266 - 21:40:44.648828] [ 26.09%] | | | |__det_model(0.3596)
[2022-04-15 21:40:44.314523 - 21:40:44.638856] [ 23.53%] | | | | |__paddle_run(0.3243)
[2022-04-15 21:40:44.661930 - 21:40:44.740922] [  5.73%] | | | | |__cls_model(0.0790)
[2022-04-15 21:40:44.662881 - 21:40:44.707690] [  3.25%] | | | | | |__paddle_run(0.0448)
[2022-04-15 21:40:44.708402 - 21:40:44.740841] [  2.35%] | | | | | | |__paddle_run(0.0324)
[2022-04-15 21:40:44.741082 - 21:40:45.273045] [ 38.60%] | | | | |__rec_model(0.5320)
[2022-04-15 21:40:44.743606 - 21:40:45.040089] [ 21.51%] | | | | | |__paddle_run(0.2965)
[2022-04-15 21:40:45.042587 - 21:40:45.272397] [ 16.67%] | | | | | | |__paddle_run(0.2298)
```

Check <https://github.com/oracle-samples/hiq> for python source code, ipynb, and tutorial

Real Use Case - PaddleOCR Memory

```
1 import hiq
2 from hiq.framework.paddleocr import HIQ_PADDLEOCR_CONF
3 from hiq.framework.paddle import PaddleHiQMemory
4
5 driver = PaddleHiQMemory(HIQ_PADDLEOCR_CONF)
6 hiq.mod("main").main()
7 driver.show()
```

```
[ 253.863 - 578.625] [100.00%] ●_root_get_memory_mb(324.7617)
[ 253.863 - 358.316] [ 32.16%] |__init__(104.4531)
[ 253.906 - 301.051] [ 14.52%] | |__create_predictor(47.1445)
[ 301.051 - 329.965] [  8.90%] | |__create_predictor(28.9141)
[ 329.965 - 358.316] [  8.73%] | |__create_predictor(28.3516)
[ 358.316 - 578.625] [ 67.84%] |__ocr(220.3086)
[ 364.438 - 583.648] [ 67.50%] |__txt_model(219.2109)
[ 368.438 - 579.031] [ 64.85%] |__det_model(210.5938)
[ 394.723 - 596.930] [ 62.26%] | |__paddle_run(202.2070)
[ 579.031 - 583.648] [  1.42%] |__rec_model(4.6172)
[ 583.098 - 583.648] [  0.17%] |__paddle_run(0.5508)
```

Check <https://github.com/oracle-samples/hiq> for python source code, ipynb, and tutorial

Documents and Tutorial About HiQ



- Online Document: <https://hiq.readthedocs.io/en/latest/index.html>
- User Guide: <https://github.com/oracle-samples/hiq/blob/main/hiq/docs/hiq.pdf>
- Logging: https://hiq.readthedocs.io/en/latest/4_o_advanced.html#log-monkey-king
- Tracing: https://hiq.readthedocs.io/en/latest/5_distributed.html
 - Zipkin: https://hiq.readthedocs.io/en/latest/5_distributed.html#zipkin
 - Jaeger: https://hiq.readthedocs.io/en/latest/5_distributed.html#jaeger
- Metrics:
 - Prometheus: https://hiq.readthedocs.io/en/latest/7_integration.html#prometheus
- Streaming:
 - Kafka: https://hiq.readthedocs.io/en/latest/7_integration.html#oci-streaming

Welcome to contribute!



Future of Observability

- Unification (logging, tracing, monitoring, alerting, debugging...)
- Automation
- Standardization



<https://github.com/oracle-samples/hig>

