Tracing Kubernetes

Benjamin Saint-Cyr

Polytechnique Montréal

DORSAL Laboratory

Agenda

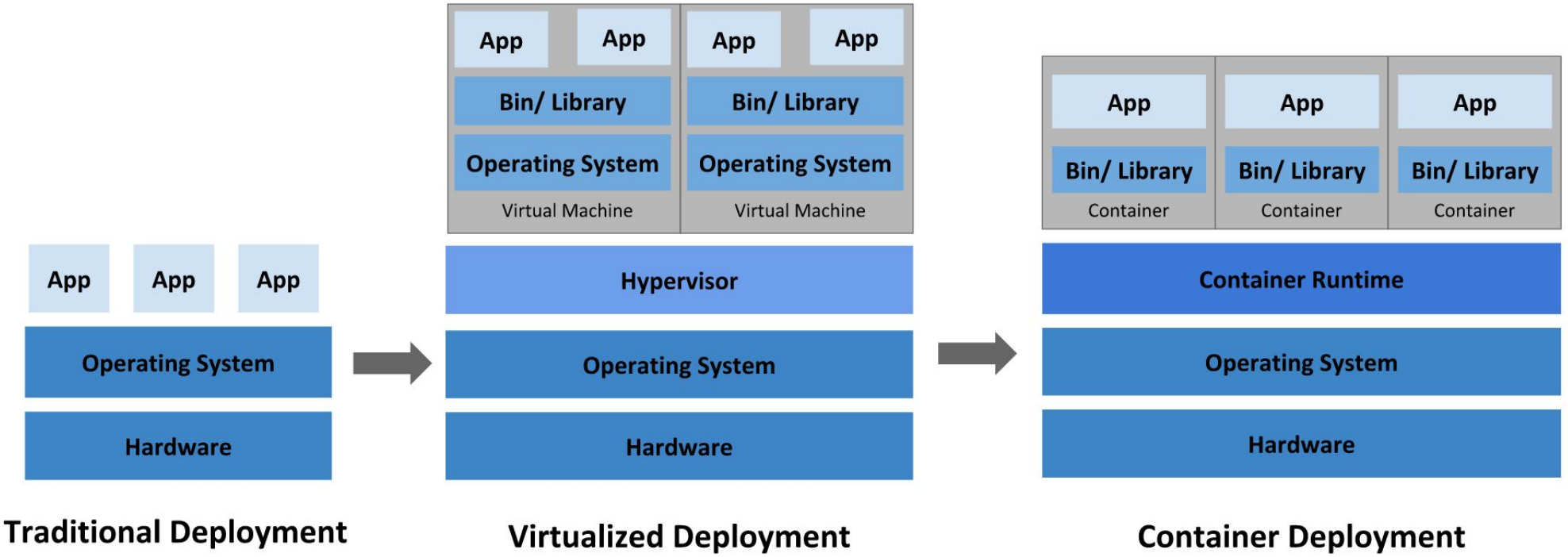
- Context
- Methodology
- Results
- Conclusion

Agenda

- Context
- Methodology
- Results
- Conclusion

Context

The evolution of Deployments



Context

- Kubernetes: Industry Standard for Container Orchestration
  - Manages container lifecycle
  - Provides auto-scaling, load balancing, volume management
  - Declarative configuration
  - Robust ecosystem and strong community support

Context

- Pod efficiency is great to optimise cloud computing costs
- Fast pod startup allows for better elastic scaling
- The demand for better scalability is increasing
  - Kubernetes supports up to 5,000 nodes
  - OpenAi is using 7,500 nodes
  - Alibaba uses 10,000 nodes (largest cluster)

Agenda

- Context
- Methodology
- Results
- Conclusion

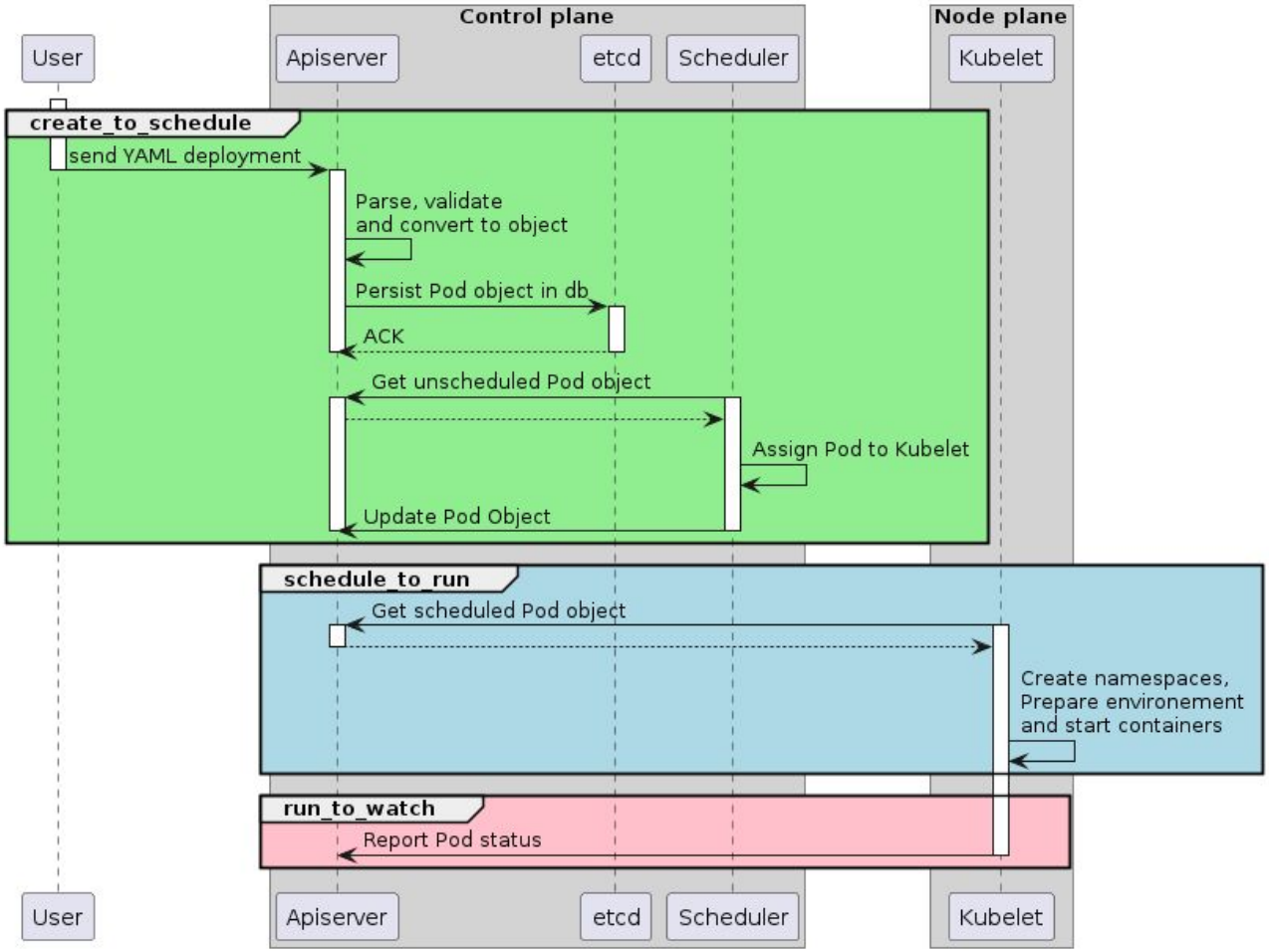
Methodology

LTTng-UST in Go using CGO (Source)

```go
package lttng

/*
#cgo LDFLAGS: -ldl -llttng-ust

#define TRACEPOINT_DEFINE
#include "k8s-tp.h"

void traceStartSpan(uint64_t s_id, uint64_t s_p_id, char* o_name, char* o_ctx) {
        tracepoint(k8s_ust, start_span, s_id, s_p_id, o_name, o_ctx);
}

void traceEndSpan(uint64_t s_id, char* o_ctx) {
        tracepoint(k8s_ust, end_span, s_id, o_ctx);
}
*/
import "C"

func ReportStartSpan(spanID uint64, parentID uint64, operationName string, context string) {
        C.traceStartSpan(
                C.uint64_t(spanID),
                C.uint64_t(parentID),
                C.CString(operationName),
                C.CString(context),
        )
}

func ReportEndSpan(spanID uint64, context string) {
        C.traceEndSpan(
                C.uint64_t(spanID),
                C.CString(context),
        )
}
```

Methodology

The pod startup process

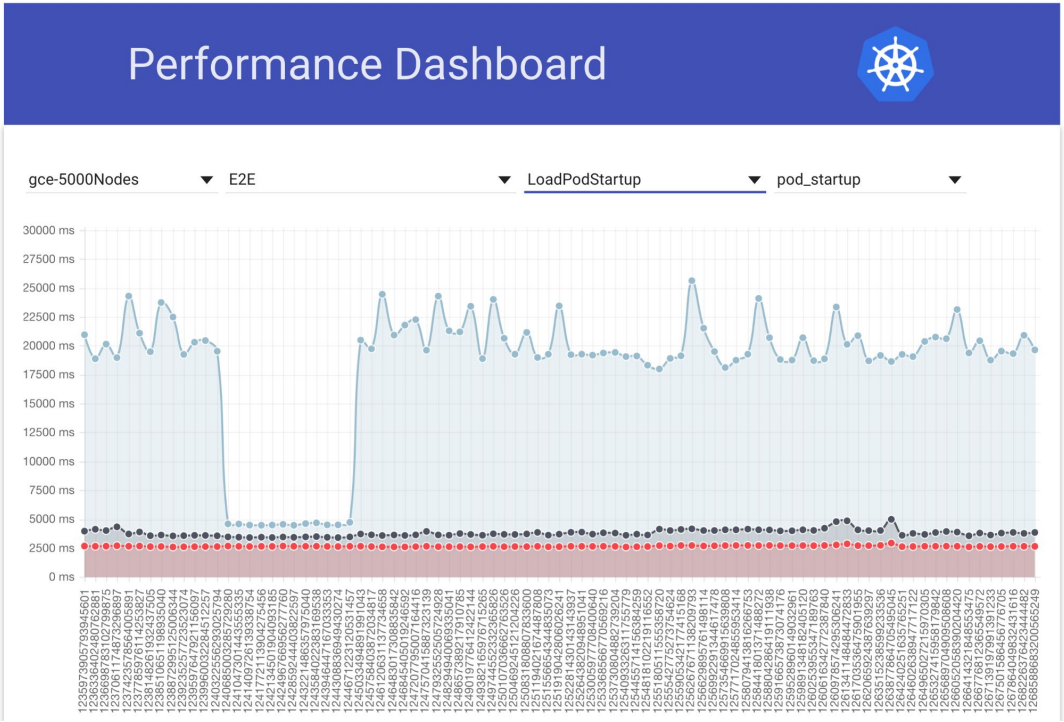


Agenda

- Context
- Methodology
- Results
- Conclusion

Common issues

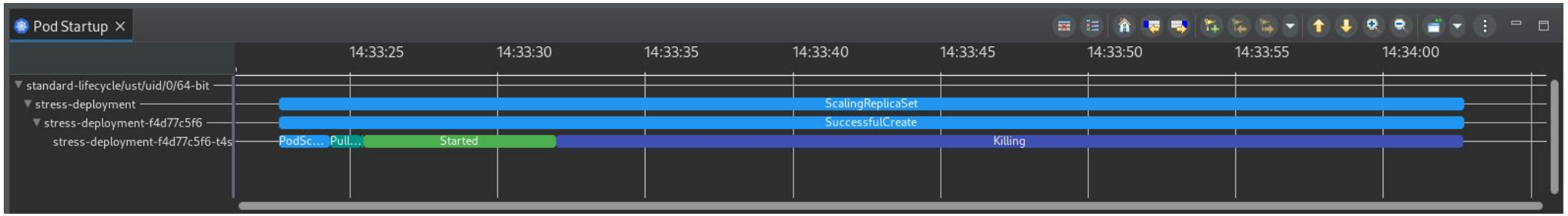
- The maintainers of kubernetes also have some issues that could be solved by kernel tracing
- Unknown pod startup latency during load testing
- Common Scalability Issues



Results

3. Visualisation in Trace compass: Life Cycle Analysis

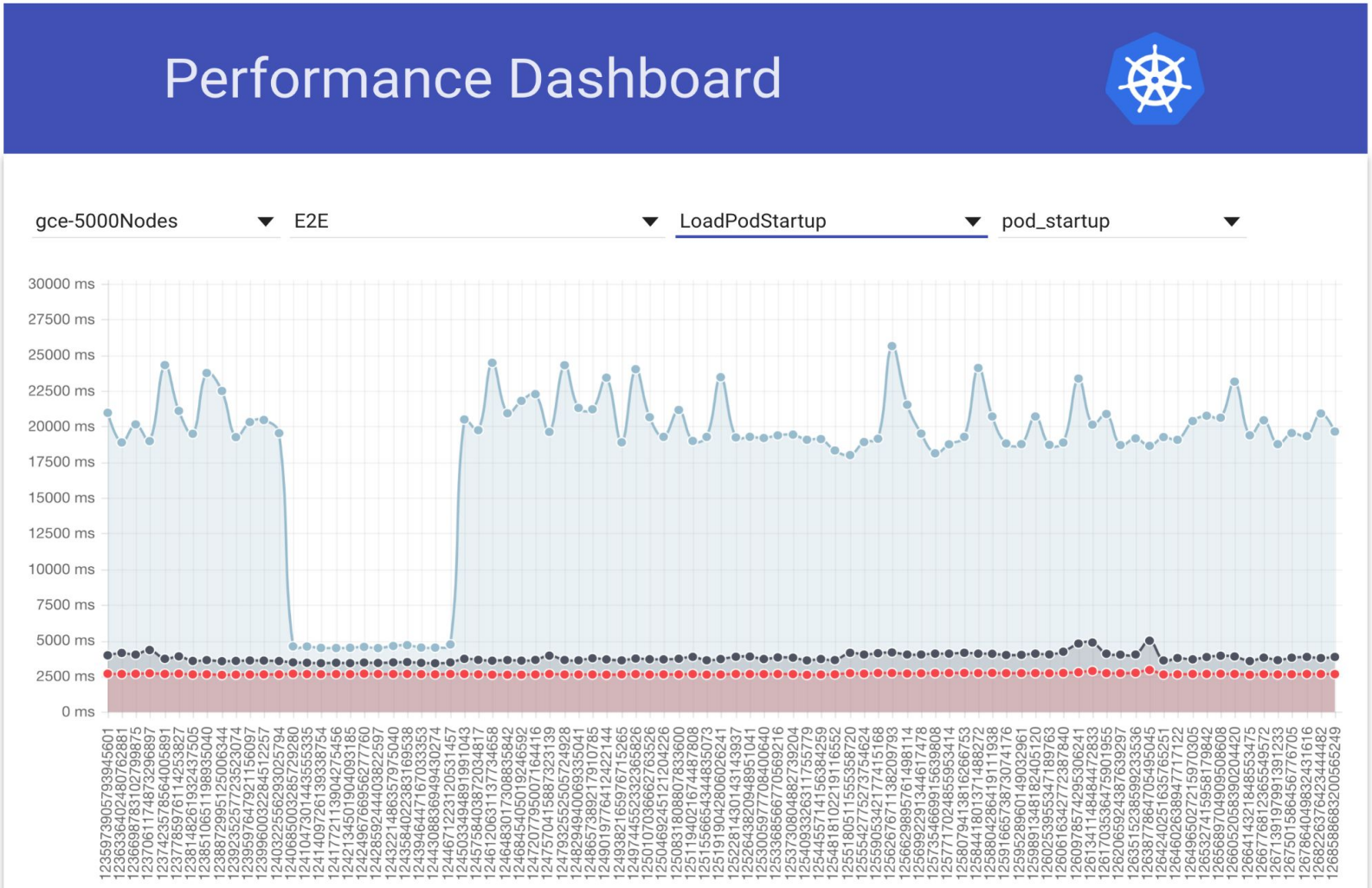
Standard Pod Lifecycle



Methodology

3. Cluster loader 2 and perf dash

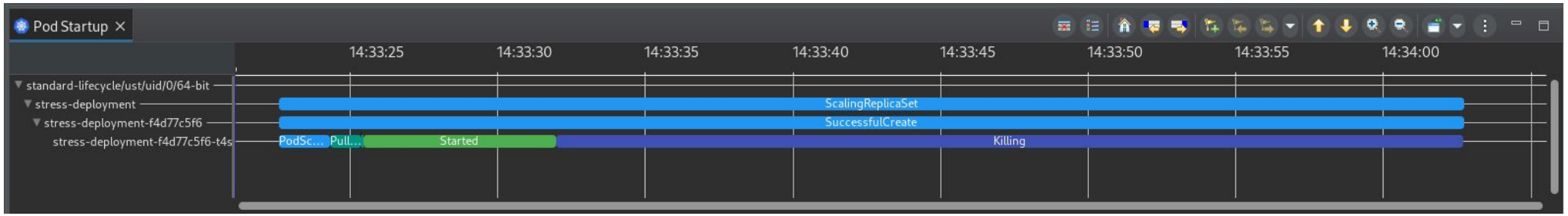

```
PodStartupLatency_PodStartupLatency: {
  "version": "1.0",
  "dataItems": [
    {
      "data": {
        "Perc50": 6126.807172,
        "Perc90": 7720.925038,
        "Perc99": 8165.909724
      },
      "unit": "ms",
      "labels": {
        "Metric": "schedule_to_watch"
      }
    },
    {
      "data": {
        "Perc50": 6874.925282,
        "Perc90": 8471.302985,
        "Perc99": 8868.321156
      },
      "unit": "ms",
      "labels": {
        "Metric": "pod_startup"
      }
    },
    {
      "data": {
        "Perc50": 748.11811,
        "Perc90": 776.917592,
        "Perc99": 782.059061
      },
      "unit": "ms",
      "labels": {
        "Metric": "create_to_schedule"
      }
    },
    {
      "data": {
        "Perc50": 3226.426988,
        "Perc90": 4249.622053,
        "Perc99": 4297.588568
      },
      "unit": "ms",
      "labels": {
        "Metric": "schedule_to_run"
      }
    },
    {
      "data": {
        "Perc50": 3067.431759,
        "Perc90": 3677.860045,
        "Perc99": 3868.321156
      },
      "unit": "ms",
      "labels": {
        "Metric": "run_to_watch"
      }
    }
  ]
}
```

Results

3. Visualisation in Trace compass: Life Cycle Analysis

Standard Pod Lifecycle



Unsuccessful Pod Startup

Results

3. Visualisation in Trace compass: Cgroup CPU Usage

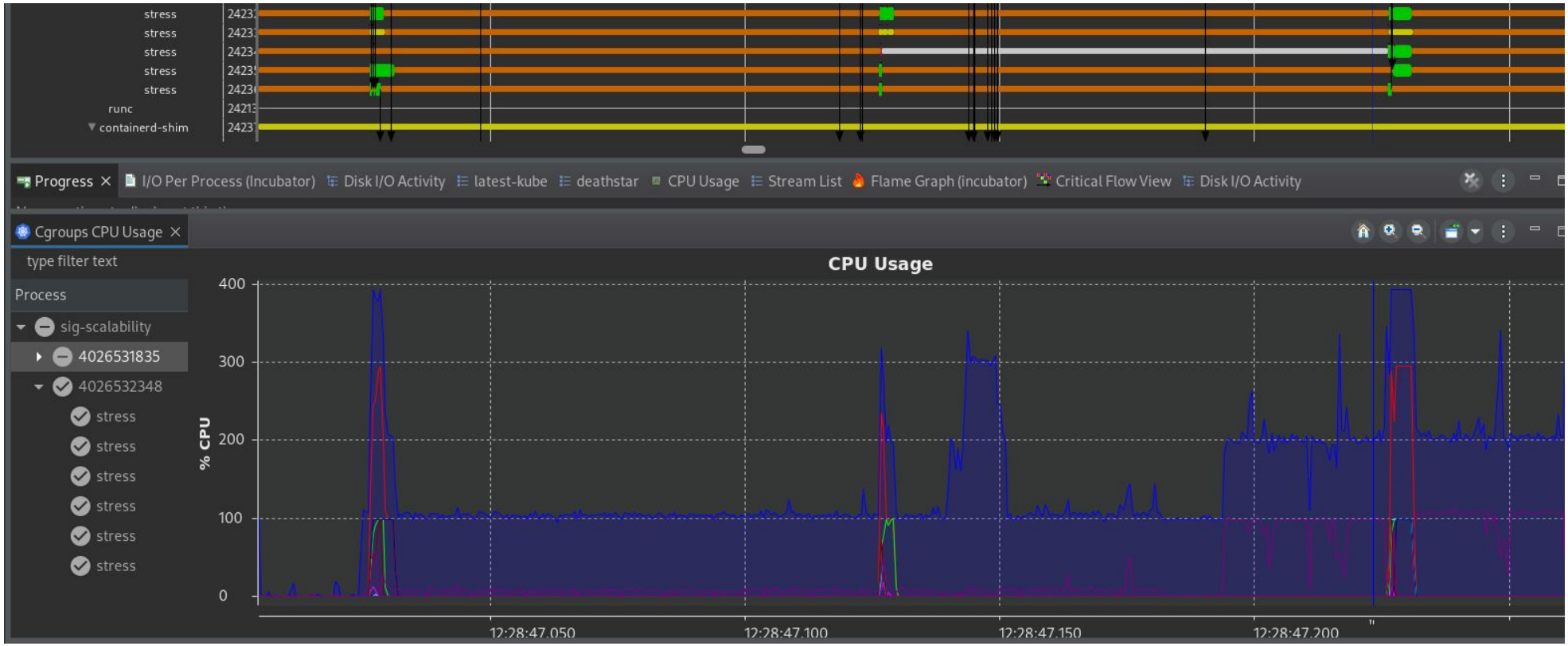
Common issues

- One common issue comes from CPU limits
- Latency due to low CPU Limits

The app needed 100ms
The app was throttled 120ms

Results

3. Visualisation in Trace compass: Cgroup CPU Usage



Agenda

- Context
- Methodology
- Results
- Conclusion

Conclusion

- Using kernel tracing provides deeper insight into the performance of kubernetes
- This methodology can be applied to analyze other container runtime and orchestrator combinations
- Incorporate Go runtime tracing in future work

Conclusion

Thank you!

Questions?

References

9 Insights on Real-World Container Use | Datadog. (n.d.). Retrieved May 29, 2023, from https://www.datadoghq.com/container-report/

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. *Communications of the ACM*, *59*(5), 50–57. https://doi.org/10.1145/2890784

Considerations for large clusters. (n.d.). Kubernetes. Retrieved May 29, 2023, from https://kubernetes.io/docs/setup/best-practices/cluster-large/

How Does Alibaba Ensure the Performance of System Components in a 10,000-node Kubernetes Cluster? - Alibaba Cloud Community. (n.d.). Retrieved May 29, 2023, from https://www.alibabacloud.com/blog/how-does-alibaba-ensure-the-performance-of-system-components-in-a-10000-node-kubernetes-cluster_595469

KubeCon + CloudNativeCon | Open Source Summit China 2019: Understanding Scalability and Performanc... (n.d.). Retrieved May 29, 2023, from https://kccncosschn19eng.sched.com/event/Nroo

Kubernetes in the wild report 2023. (n.d.). Retrieved May 29, 2023, from https://www.dynatrace.com/news/blog/kubernetes-in-the-wild-2023/

Manaouil, K., & Lebre, A. (2020). *Kubernetes and the Edge?* (p. 19) [Report, Inria Rennes - Bretagne Atlantique]. https://hal.inria.fr/hal-02972686

Medel, V., Tolosana-Calasanz, R., Bañares, J. Á., Arronategui, U., & Rana, O. F. (2018). Characterising resource management performance in Kubernetes. *Computers & Electrical Engineering*, *68*, 286–297. https://doi.org/10.1016/j.compeleceng.2018.03.041

Overview. (n.d.). Kubernetes. Retrieved May 30, 2023, from https://kubernetes.io/docs/concepts/overview/

Scaling Kubernetes to 7,500 nodes. (n.d.). Retrieved May 29, 2023, from https://openai.com/research/scaling-kubernetes-to-7500-nodes

Using Prometheus to Avoid Disasters with Kubernetes CPU Limits | Containers. (n.d.). Retrieved May 29, 2023, from https://aws.amazon.com/blogs/containers/using-prometheus-to-avoid-disasters-with-kubernetes-cpu-limits/