Observability at Expedia

By Ayan Sen, Vinay Sen

Observability events

- Logs
  - Stateless events
  - Collected by local agent, pushed to the backend (Elastic Search, Some DB etc..
- Metrics
  - Slightly more sophisticated than logs.
- Traces
  - Span - represents a service call or block of code.
  - Trace - collection of spans correlated by an identifier

Distributed Tracing

Reference: Dapper - paper on distributed tracing from Google

- Much needed in a micro services architecture
- Existing solutions
  - ZipKin (Twitter)
  - X-Ray (Amazon)
  - Jaeger (Uber)
  - Haystack (Expedia)

Haystack Architecture

- Uses Kafka at its core
- Reads data from Kafka, computes the metrics,
- Subsystems
  - Traces
    - Uses both ElasticSearch and Cassandra in the backend.
  - Trends
    - Transformer - reads data stream from Kafka and calculate trends
    - Aggregator - feeds off the output from Transformer, and generate some metrics.
    - For metrics, uses MetricTank -

    - Service Graph
        - Kafka apps: node-finder, graph-builder
        - Aka k-stream apps.
    - Anamoly Detection
    - Pipes

Haystack at Expedia

- In use at multiple brands of Expedia
- 400K spans/sec ingestion
- 50 node C5.xlarge Casssandra
- 65+ node C5.xlarge k8s cluster - what is c5.xlarge?
- All the micro services and sub components run in Kubernetes containers
- Supports OpenTracing clients in Java, odeJS, Go.. Python coming soon.
- Zipkin to Haystack span converter
- Deployment done through Terraform scripts
- Actively looking for contributors

References

- https://github.com/ExpediaDotCom/haystack
- http://bit.ly/ato/haystack-io (link seem to be not working)