Trace tuning

Data Prepper for trace analytics in version 0.8.x supports both vertical and horizontal scaling. Adjust the size of your single Data Prepper instance to meet your workload's demands to scale vertically.

Scale horizontally by deploying multiple Data Prepper instances to form a cluster by using the Core Peer Forwarder. This enables Data Prepper instances to communicate with instances in the cluster, and is required for horizontally-scaling deployments.

Scaling tips

Below are some useful tips for scaling. We recommend that you modify parameters based on the requirements. Also, monitor the Data Prepper host metrics and OpenSearch metrics to ensure the configuration is working as expected.

Buffer

The total number of trace requests that Data Prepper is processing is equal to sum of `buffer_size` in `otel-trace-pipeline` and `raw-trace-pipeline`.

The total number of trace requests inflight to OpenSearch is equal to the product of `batch_size` and `workers` in `raw-trace-pipeline`.

We recommend:

- have the same `buffer_size` value in `otel-trace-pipeline` and `raw-trace-pipeline`
- `buffer_size` >= `workers` * `batch_size` in the `raw-trace-pipeline`

Workers

The `workers` setting determines the number of threads that will be used by Data Prepper to process requests from the buffer.

We recommend that you set the workers based on the CPU utilization. This value can be higher than available processors as the Data Prepper spends significant I/O time in sending data to OpenSearch.

Heap

Configure the heap of Data Prepper by setting the `JVM_OPTS` environmental variable.

We recommend that you set the heap value to a minimum of `4` * `batch_size` * `otel_send_batch_size` * `maximum size of indvidual span`.

As mentioned in the setup, set `otel_send_batch_size` as `50` in your opentelemetry collector configuration.

Disk

Because Data Prepper uses disk to store metadata required for service-map processing, we store only key fields `traceId`, `spanId`, `parentSpanId`, `spanKind`, `spanName` and `serviceName`. The service-map

plugin ensures it only stores two files with each storing `window_duration` seconds of data. During testing, we found that for a throughput of `3000 spans/second`, the total disk usages was `4 MB`.

Data Prepper uses the disk to write logs. In the current version, you can redirect the logs to the path of your preference.

AWS

AWS EC2 Cloudformation template provides user-friendly mechanism to configure the above scaling attributes.

Kubernetes config files and EKS config files are available to configure these attributes in a cluster deployment.

Benchmark

We ran tests in a `r5.xlarge` with the below configuration,

- `buffer_size` : `4096`
- `batch_size` : `256`
- `workers` : 8
- `Heap` : 10GB

The above setup was able to handle a throughput of `2100` spans/second at `20` percent CPU utilization.