OpenSearchCon INDIA

PPL Power-Up
Advanced Data Transformation Pipelines for Observability

OpenSearchCon Mumbai

The 2AM Problem

Microservice based architecture , PagerDuty fire , find the root cause from hundreds of microservices.

Finding root cause
Step 1 : Find services with error spikes in the last hour
Step 2: Correlate trace_ids with upstream spans.
Step 3 : Join with a service ownership table.

DSL

Step 1

```
POST /application_logs/_search
{"size": 0,
 "query":{"bool":{"filter":[
  {"range":{"@timestamp":{"gte":"now-
       1h"}}},
  {"range":{"status":{"gte":500}}}
 ]}},
 "aggs":{"by_service":{
  "terms":{"field":"service.name"},
  "aggs":{"by_trace":{
   "terms":{"field":"trace_id"}
  }}
 }}
}
```

Find Error count by service, trace id

Step 2

```
POST /distributed_traces/_search
{"size": 0,
  "query":{"bool":{"filter":[
   {"terms":{"trace_id":["trace-
      001","trace-002",...]}},
   {"term":{"kind":"CLIENT"}}
  ]}},
  "aggs":{"by_upstream":{
      "terms":{"field":"upstream_serv
      ice"},
      "aggs":{"avg_duration":{"avg":{
      "field":"duration_ms"}},

      "failure_count":{"filter":{"range
      ":{"status_code":{"gte":500}}}}}
   }}
  }
```

Lookup Upstream services

Step 3

```
POST /service_catalog/_search
{"query":{"terms":{"service_name":["
      api-gateway","order-svc"]}},

      "_source":["service_name","tea
      m","oncall_engineer","slack_cha
      nnel"]
}
```

Look up service ownership from service_catalog

What if: 1 query , just 1 round trip , cross-index join built-in and human-readable?

```
source = application_logs | where @timestamp  > now() - interval 1 hour
    AND status >= 500 | stats count() as errors  by service, trace_id | where errors > 10
| join left=l right=r ON l.trace_id = r.trace_id  [source = distributed_traces   | fields
      trace_id, upstream_service]
| lookup service_owners service  replace owner, team, pager | sort -affected_traces
```

OpenSearchCon INDIA

PPL Power-Up
Advanced Data Transformation Pipelines for Observability

Bharav Patel

Specialist SA, OpenSearch

AWS

Three Query Languages

DSL

PPL

SQL

Log Investigation + Correlation

OpenSearchCon INDIA

Problem

Alerts fire for payment-svc. You need to find: Which service is failing? Who is calling it? Who owns it? When did the spike start? Today this requires 3 separate queries + Python glue code.

What we'll find

payment-svc has errors. api-gateway is making failing calls averaging 2.8s. Owner: @alice in #platform-ops. Spike started at a specific 5-minute window. All from ONE pipeline.

Data we're working with

application_logs

distributed_traces

service_catalog

OpenSearch Dashboards

Discover

a

Update

PPL

```
1
```

1 line Completed in 533 ms Recent queries

Results (4) Download as CSV

_source

> total: 21 errors: 1 minute: 2026-05-21T08:00:00.000+00:00 rolling_avg: 0.3333333333333333 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

> total: 20 errors: 2 minute: 2026-05-21T08:25:00.000+00:00 rolling_avg: 0.6666666666666666 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

> total: 22 errors: 1 minute: 2026-05-21T08:50:00.000+00:00 rolling_avg: 0.3333333333333333 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

> total: 69 errors: 47 minute: 2026-05-21T09:00:00.000+00:00 rolling_avg: 16 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

Multivalue Field Operations

OpenSearchCon INDIA

Problem

Each error event is tagged with multiple failure patterns like ['timeout', 'circuit-breaker', 'payment']. You need to count individual tags, filter to specific ones, and see service paths.

What we'll find

'high-priority' is the most common tag. checkout-ui has the most timeout errors. Each trace touches 2-3 services visible in a single collapsed row.

Data we're working with

api_logs — API gateway events — each tagged with 1-4 failure patterns stored as arrays (e.g. timeout, circuit-breaker, payment)

OpenSearch Dashboards

Discover

a

Update

service_catalog

Search field names 0

Selected fields

_source

Available fields

_id

_index

_score

_type

anomaly

errors

minute

rolling_avg

total

PPL

```
1
```

1 line Completed in 178 ms Recent queries

Results (4) Download as CSV

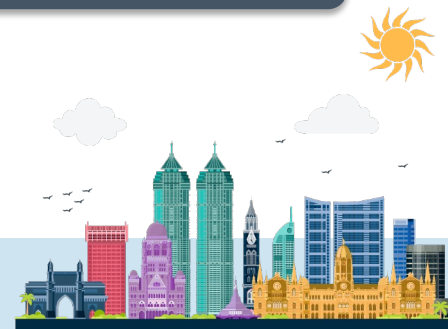
_source
total: 21 errors: 1 minute: 2026-05-21T08:00:00.000+00:00 rolling_avg: 0.3333333333333333 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -
total: 20 errors: 2 minute: 2026-05-21T08:25:00.000+00:00 rolling_avg: 0.6666666666666666 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -
total: 22 errors: 1 minute: 2026-05-21T08:50:00.000+00:00 rolling_avg: 0.3333333333333333 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -
total: 69 errors: 47 minute: 2026-05-21T09:00:00.000+00:00 rolling_avg: 16 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

Apache Calcite — The Engine Behind PPL

- Apache Hive
- Apache Flink
- Apache Druid
- Apache Drill
- OpenSearch PPL

Apache Calcite

- Relational Algebra
  Formal query model
- Query Optimizer
  Rule-based pushdown
- Type System
  SQL-compatible
- Adapter Framework
  Pluggable backends



How Queries Execute: DSL vs PPL

DSL Path

JSON Query

Coordinator (scatter to shards)

Lucene IndexSearcher + Aggregator

Reduce + Fetch (merge shards)

Result (shard-level cached)

PPL Path

PPL Text -> ANTLR Parser -> AST

CalciteRelNodeVisitor -> Relational Algebra

Calcite Optimizer (pushdown rules)

Planning

Pushed to OpenSearch (where, stats, sort)

Calcite In-Memory (join, streamstats, mv*)

Combined Result

Pushdown: What Runs Where

source=logs | where status>=500 | stats count() by svc | join ... | streamstats ...

Calcite Optimizer

PUSHED TO OPENSEARCH (Lucene speed)

where -> DSL bool filter
fields -> _source includes
stats count/avg/sum -> DSL aggregations
sort -> DSL sort builders
head -> DSL size parameter
highlight -> DSL HighlightBuilder

CALCITE IN-MEMORY (new capabilities)

join / lookup -> MergeJoin / HashJoin
streamstats -> Window functions
mvcombine / mvexpand -> Array UDFs
rex / regex -> Text extraction
chart (pivot) -> Relational transforms
appendpipe -> Union of pipelines

PPL Command Categories

CROSS-INDEX: join, lookup, multisearch

MULTIVALUE: mvcombine, mvexpand, mvmap, mvfind

AGGREGATION: stats, eventstats, streamstats, chart, addtotals, appendpipe

TRANSFORMATION: eval, parse, rex, rename, replace, convert

FILTERING: where, fields, dedup, head, regex, top, rare

DATA RETRIEVAL: source, search, describe

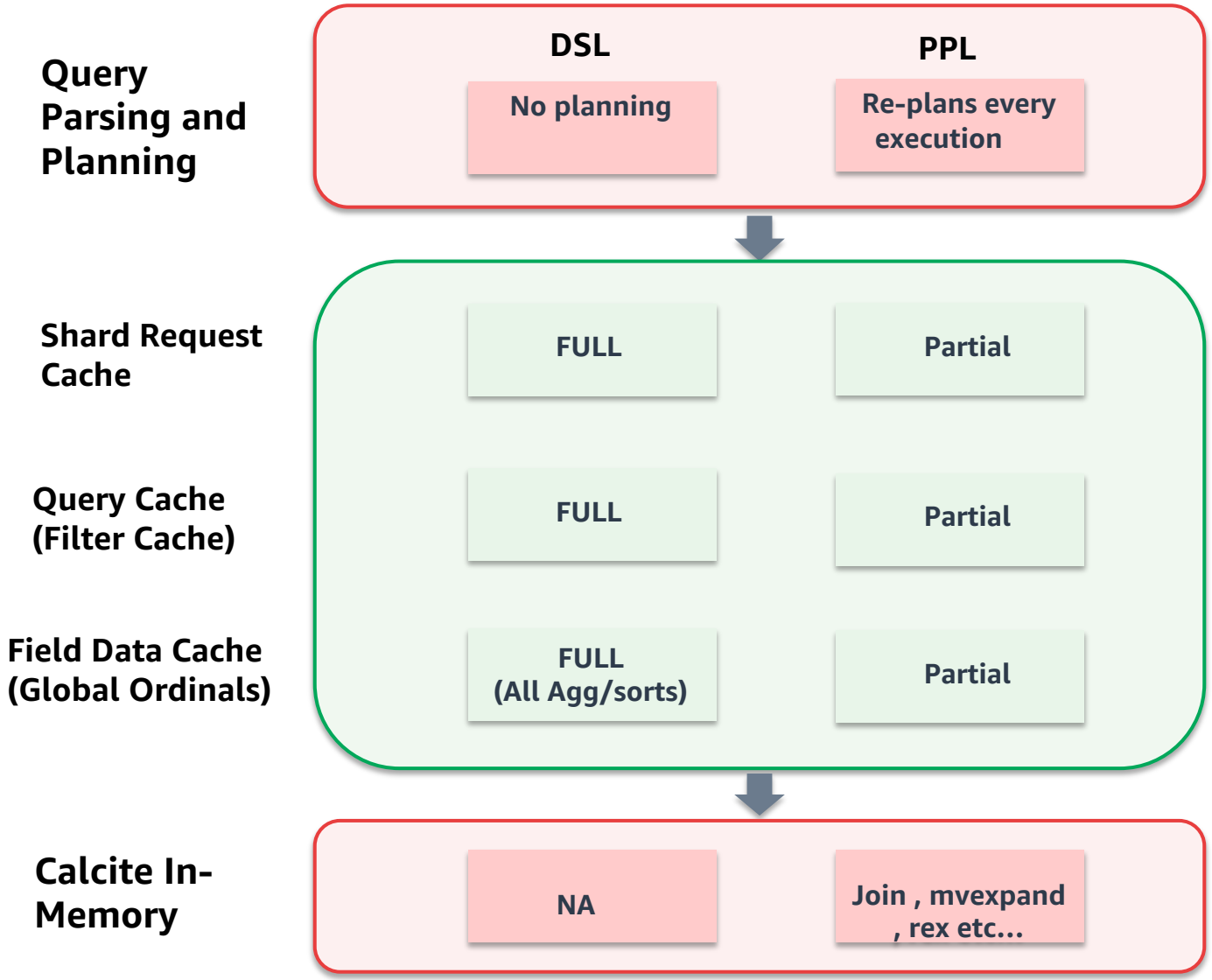
PPL Functions

OpenSearchCon INDIA

Category	PPL Example
IP Address	cidrmatch(ip, '10.0.0.0/8') geoip(client_ip)
JSON	json_extract(field, '$.error.code') json_keys(payload)
Conditional	case(status>=500, 'critical', status>=400, 'warn', else 'ok')
Full-Text	match(message, 'timeout') query_string('svc:pay* AND 500')
Aggregation	percentile(latency, 95) distinct_count_approx(user_id) earliest(message) latest(message)
Crypto	md5(field) sha2(field, 256)

+ 120 standard functions

Performance Trade-offs



When to Use DSL vs PPL

DSL

Dashboard panels (auto-refresh) — shard cache, zero overhead

Alert rule evaluation — cacheable

Programmatic APIs — all clients have native search

Complex search queries

PPL

Ad-hoc incident investigation — sequential, single round-trip

Cross-service correlation — join/lookup (no DSL equivalent)

Rolling anomaly detection — streamstats (no DSL equivalent)

Correlation-based alerting — cross-index triggers

Connect external data sources – Prometheus, S3(via SQL/PPL engine)

Coming from Splunk?

OpenSearchCon INDIA

SPL Commands

Stats
Streamstats
Mvexpand
Mvcombine
Mvzip
Rex
Lookup
Join
Chart
Eval
Dedup
Appendpipe

PPL Commands

Stats
Streamstats
Mvexpand
Mvcombine
Mvzip
Rex
Lookup
Join
Chart
Eval
Dedup
Appendpipe

Production Settings & Tips

OpenSearchCon INDIA

Key settings:

plugins.calcite.pushdown.enabled: true

plugins.calcite.fallback.allowed: true

plugins.ppl.join.subsearch_maxout: 50000

query:enhancements:enabled: true (Dashboards setting — in opensearch_dashboards.yml)

Tips:

Add head N early — limits data through Calcite

Use lookup over join for static tables

Key Takeaways

- PPL makes investigation easy
- Hybrid execution = best of both worlds
- Use the right tool for the job
- Start with top queries

Resources

OpenSearchCon INDIA

Documentation

https://docs.opensearch.org/latest/sql-and-ppl/ppl/index/

GitHub

https://github.com/opensearch-project/piped-processing-language

Community

forum.opensearch.org | opensearch.slack.com

OpenSearchCon INDIA

Thank You!

Questions?

OpenSearch Dashboards

Discover

a

Update

PPL

```
1
```

1 line Completed in 533 ms Recent queries

Results (4) Download as CSV

_source

> total: 21 errors: 1 minute: 2026-05-21T08:00:00.000+00:00 rolling_avg: 0.3333333333333333 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

> total: 20 errors: 2 minute: 2026-05-21T08:25:00.000+00:00 rolling_avg: 0.6666666666666666 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

> total: 22 errors: 1 minute: 2026-05-21T08:50:00.000+00:00 rolling_avg: 0.3333333333333333 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

> total: 69 errors: 47 minute: 2026-05-21T09:00:00.000+00:00 rolling_avg: 16 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -

OpenSearch Dashboards

Discover

a

Update

service_catalog

Search field names 0

Selected fields

_source

Available fields

_id

_index

_score

_type

anomaly

errors

minute

rolling_avg

total

PPL

```
1
```

1 line Completed in 178 ms Recent queries

Results (4) Download as CSV

_source
total: 21 errors: 1 minute: 2026-05-21T08:00:00.000+00:00 rolling_avg: 0.3333333333333333 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -
total: 20 errors: 2 minute: 2026-05-21T08:25:00.000+00:00 rolling_avg: 0.6666666666666666 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -
total: 22 errors: 1 minute: 2026-05-21T08:50:00.000+00:00 rolling_avg: 0.3333333333333333 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -
total: 69 errors: 47 minute: 2026-05-21T09:00:00.000+00:00 rolling_avg: 16 anomaly: true _id: - _type: - _index: 7f37b960-54dd-11f1-9658-8f5a7247f26f _score: -