OpenSearchCon INDIA

# The Secret Life of a Search Query

## A Fun, Visual Journey Through How OpenSearch Really Thinks

Views expressed are my own and do not represent those of my organization.

# Meet Your Tour Guide – Shubhi Khanna




## My qualifications for today's talk:

- I like distributed systems.
- I ask too many "but what actually happens?" questions.
- I went down the OpenSearch rabbit hole, so you don't have to.

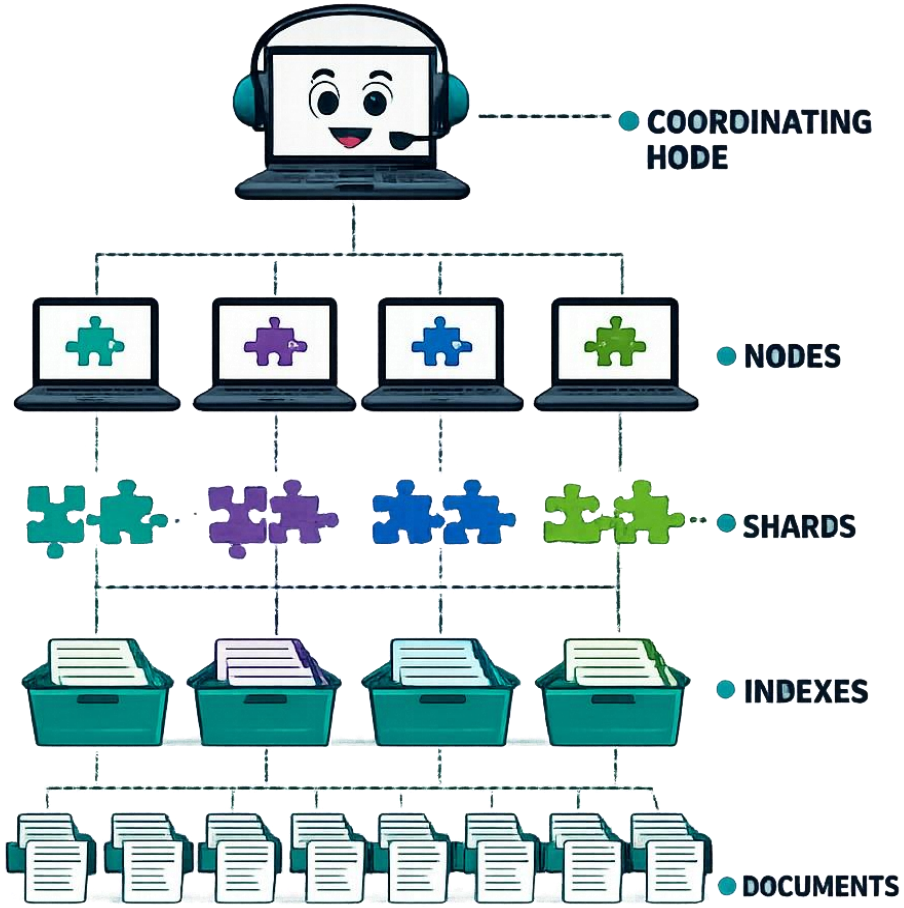
# One Tiny Search Box, One Giant Adventure

## What happens inside OpenSearch after a query is sent to the cluster

# Meet the Cast

OpenSearchCon INDIA

**Documents, indexes, nodes, shards, segments, and the coordinating node that keeps the chaos organized**


# Analyzer Spa Day



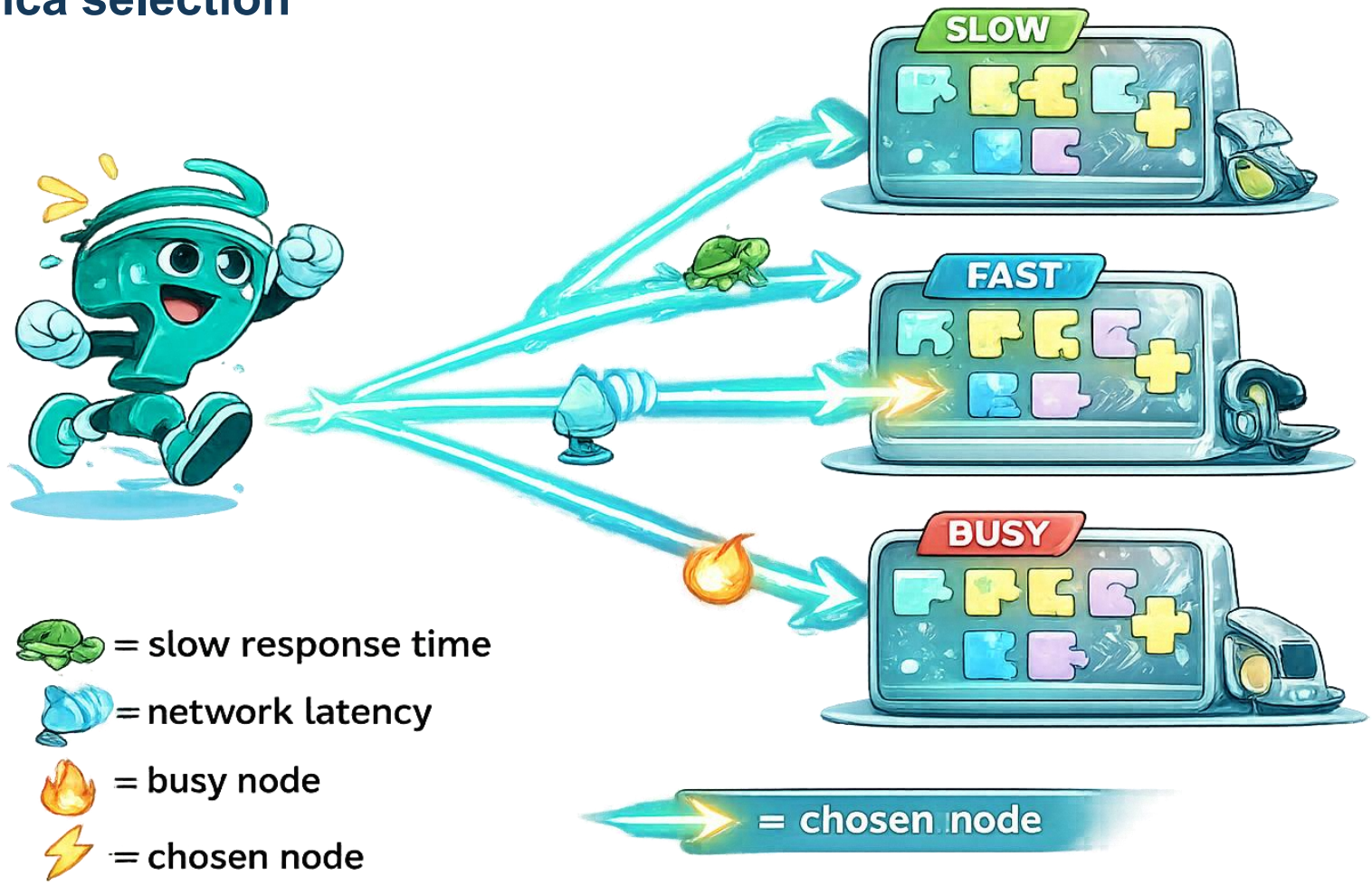
**How OpenSearch uses character filters, a tokenizer, and token filters to turn raw text into searchable terms**

| | |
|---|---|
| Raw Text | Running in THE park!!! |
| Character Filter | Running in THE park |
| Tokenizer | ["Running", "in", "THE", "park"] |
| Token Filters | ["run", "park"] |
| Final Terms | Terms stored in index: run, park |

# Shard Hopscotch

OpenSearchCon INDIA

## How search shard routing sends a query to primary or replica shards using adaptive replica selection




**Adaptive replica selection**
based on:

- Previous response time
- Network latency
- Queue size

# Auditions, Then Callback

OpenSearchCon INDIA

## How the query phase finds and scores matches locally, and the fetch phase retrieves the final documents

### Query Phase


### Fetch Phase

# Backstage Wizards

OpenSearchCon INDIA

## How search pipelines use request processors, response processors, and phase results processors on the coordinating node

1. Request Processors
2. Phase Results Processors
3. Response Processors

**Search pipelines - A sequence of processors that can operate on your query or results**

```
Query → [Request Processors] → Search Engine → [Response Processors] → Results
                ↘ (Phase Results Processors between phases)
```

# When Keywords Meet Vibes

OpenSearchCon INDIA

**How vector search adds semantic similarity with embeddings, k-NN search, and neural queries**

### Keyword search

Query: "cheap running shoes“

Results:
✅ “cheap running shoes”
❌ misses “affordable sneakers”

### Vector / semantic search

Query:"cheap running shoes“

Results:
✅ “affordable sneakers”
✅ “budget running footwear”
✅ “low-cost jogging shoes”

Convert text into embeddings → Apply K-Nearest Neighbor algorithm

# Couples Therapy for Rankings



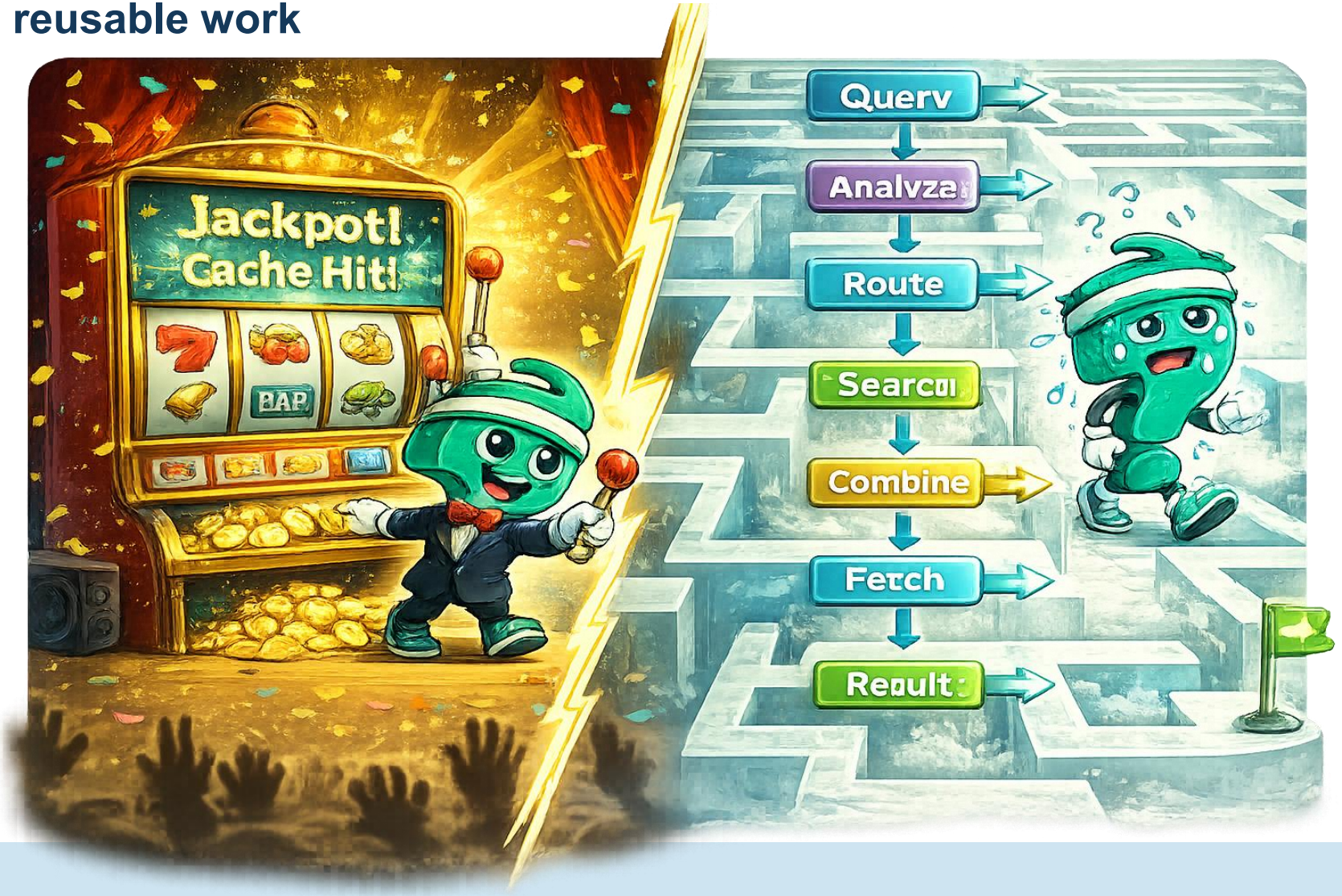
## How hybrid search combines lexical and semantic results using score normalization in a search pipeline


Score Normalization → Score Combination → Weighted Average

# Cache Me If You Can

OpenSearchCon INDIA

## How request cache, query cache, and field data cache can reduce latency for repeated or reusable work




### The 3 cache types

1. Request Cache - Remember the full answer
2. Query Cache - Reuse parts of the work
3. Field Data Cache - Precompute expensive stuff”

# Turbo Mode, But Read the Fine Print

OpenSearchCon INDIA

**How query rewriting and concurrent segment search can improve performance, sometimes with extra CPU cost**

## Query Rewriting

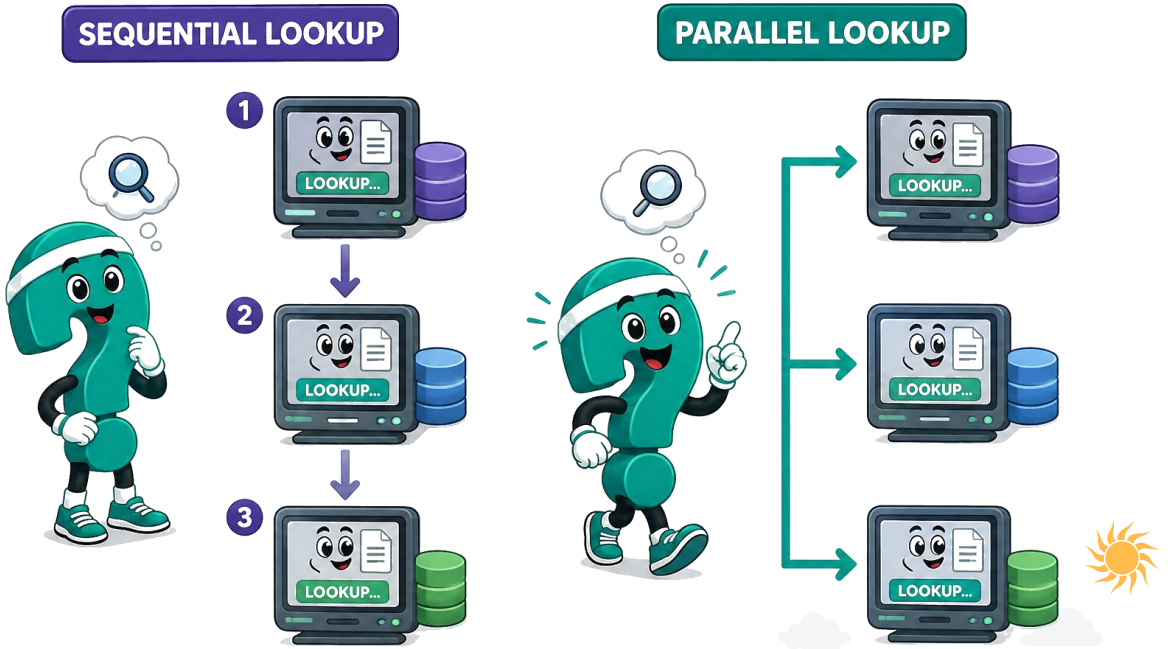
Think smarter before running

```
{
  "query": {
    "prefix": {
      "title": "run"
    }
  }
}
```

```
{
  "query": {
    "bool": {
      "should": [
        { "term": { "title": "run" } },
        { "term": { "title": "running" } },
        { "term": { "title": "runner" } },
        { "term": { "title": "runs" } }
      ]
    }
  }
}
```

## Concurrent Segment Search

Work in parallel



# The Query's Hero Shot

OpenSearchCon INDIA

OpenSearchCon
INDIA

Shubhi Khanna

Firmware @ WD