**Elastic search Training Curriculum**

Life Inside a Cluster

An Empty Cluster

Cluster Health

Add an Index

Add Failover

Scale Horizontally

Then Scale Some More

Coping with Failure

Data In, Data Out

What Is a Document?

Document Metadata

\_index

\_type

\_id

Other Metadata

Indexing a Document

Using Our Own ID

Autogenerating IDs

Retrieving a Document

Retrieving Part of a Document

Checking Whether a Document Exists

Updating a Whole Document

Creating a New Document

Deleting a Document

Dealing with Conflicts

Optimistic Concurrency Control

Using Versions from an External System

Partial Updates to Documents

Using Scripts to Make Partial Updates

Updating a Document That May Not Yet Exist

Updates and Conflicts

Retrieving Multiple Documents

Cheaper in Bulk

Don’t Repeat Yourself

How Big Is Too Big?

Distributed Document Store

Routing a Document to a Shard

How Primary and Replica Shards Interact

Creating, Indexing, and Deleting a Document

Retrieving a Document

Partial Updates to a Document

Multidocument Patterns

Why the Funny Format?

Searching—The Basic Tools

The Empty Search

hits

took

shards

timeout

Multi-index, Multitype

Pagination

Search *Lite*

The \_all Field

More Complicated Queries

Mapping and Analysis

Exact Values Versus Full Text

Inverted Index

Analysis and Analyzers

Built-in Analyzers

When Analyzers Are Used

Testing Analyzers

Specifying Analyzers

Mapping

Core Simple Field Types

Viewing the Mapping

Customizing Field Mappings

Updating a Mapping

Testing the Mapping

Complex Core Field Types

Multivalue Fields

Empty Fields

Multilevel Objects

Mapping for Inner Objects

How Inner Objects are Indexed

Arrays of Inner Objects

Full-Body Search

Empty Search

Query DSL

Structure of a Query Clause

Combining Multiple Clauses

Queries and Filters

Performance Differences

When to Use Which

Most Important Queries and Filters

term Filter

terms Filter

range Filter

exists and missing Filters

bool Filter

match\_all Query

match Query

multi\_match Query

bool Query

Combining Queries with Filters

Filtering a Query

Just a Filter

A Query as a Filter

Validating Queries

Understanding Errors

Understanding Queries

Sorting and Relevance

Sorting

Sorting by Field Values

Multilevel Sorting

Sorting on Multivalue Fields

String Sorting and Multifields

What Is Relevance?

Understanding the Score

Understanding Why a Document Matched

Fielddata

Distributed Search Execution

Query Phase

Fetch Phase

Search Options

timeout

routing

search\_type

scan and scroll

Index Management

Creating an Index

Deleting an Index

Index Settings

Configuring Analyzers

Custom Analyzers

Creating a Custom Analyzer

Types and Mappings

How Lucene Sees Documents

How Types Are Implemented

Avoiding Type Gotchas

The Root Object

Properties

Metadata: \_source Field

Metadata: \_all Field

Metadata: Document Identity

Dynamic Mapping

Customizing Dynamic Mapping

date\_detection

dynamic\_templates

Default Mapping

Reindexing Your Data

Index Aliases and Zero Downtime

Inside a Shard

Making Text Searchable

Immutability

Dynamically Updatable Indices

Deletes and Updates

Near Real-Time Search

refresh API

Making Changes Persistent

flush API

Segment Merging

optimize API

Search in Depth

Structured Search

Finding Exact Values

term Filter with Numbers

term Filter with Text

Internal Filter Operation

Combining Filters

Bool Filter

Nesting Boolean Filters

Finding Multiple Exact Values

Contains, but Does Not Equal

Equals Exactly

Ranges

Ranges on Dates

Ranges on Strings

Dealing with Null Values

exists Filter

missing Filter

exists/missing on Objects

All About Caching

Independent Filter Caching

Controlling Caching

Filter Order

Full-Text Search

Term-Based Versus Full-Text

The match Query

Index Some Data

A Single-Word Query

Multiword Queries

Improving Precision

Controlling Precision

Combining Queries

Score Calculation

Controlling Precision

How match Uses bool

Boosting Query Clauses

Controlling Analysis

Default Analyzers

Configuring Analyzers in Practice

Relevance Is Broken!

Multifield Search

Multiple Query Strings

Prioritizing Clauses

Single Query String

Know Your Data

Best Fields

dis\_max Query

Tuning Best Fields Queries

tie\_breaker

multi\_match Query

Using Wildcards in Field Names

Boosting Individual Fields

Most Fields

Multifield Mapping

Cross-fields Entity Search

A Naive Approach

Problems with the most\_fields Approach

Field-Centric Queries

Problem 1: Matching the Same Word in Multiple Fields

Problem 2: Trimming the Long Tail

Problem 3: Term Frequencies

Solution

Custom \_all Fields

cross-fields Queries

Per-Field Boosting

Exact-Value Fields

Proximity Matching

Phrase Matching

Term Positions

What Is a Phrase

Mixing It Up

Multivalue Fields

Closer Is Better

Proximity for Relevance

Improving Performance

Rescoring Results

Finding Associated Words

Producing Shingles

Multifields

Searching for Shingles

Performance

Partial Matching

Postcodes and Structured Data

prefix Query

wildcard and regexp Queries

Query-Time Search-as-You-Type

Index-Time Optimizations

Ngrams for Partial Matching

Index-Time Search-as-You-Type

Preparing the Index

Querying the Field

Edge n-grams and Postcodes

Ngrams for Compound Words

Controlling Relevance

Theory Behind Relevance Scoring

Boolean Model

Term Frequency/Inverse Document Frequency (TF/IDF)

Vector Space Mode

Lucene’s Practical Scoring Function

Query Normalization Factor

Query Coordination

Index-Time Field-Level Boosting

Query-Time Boosting

Boosting an Index

t.getBoost()

Manipulating Relevance with Query Structure

Not Quite Not

boosting Query

Ignoring TF/IDF

constant\_score Query

function\_score Query

Boosting by Popularity

modifier

factor

boost\_mode

max\_boost

Boosting Filtered Subsets

filter Versus query

functions

score\_mode

Random Scoring

The Closer, The Better

Understanding the price Clause

Scoring with Scripts

Pluggable Similarity Algorithms

Okapi BM25

Changing Similarities

Configuring BM25

Relevance Tuning Is the Last 10%

Dealing with Human Language

Getting Started with Languages

Using Language Analyzers

Configuring Language Analyzers

Pitfalls of Mixing Languages

At Index Time

At Query Time

Identifying Language

One Language per Document

Foreign Words

One Language per Field

Mixed-Language Fields

Split into Separate Fields

Analyze Multiple Times

Use n-grams

Identifying Words

standard Analyzer

standard Tokenizer

Installing the ICU Plug-in

icu\_tokenizer

Tidying Up Input Text

Tokenizing HTML

Tidying Up Punctuation

Normalizing Tokens

In That Case

You Have an Accent

Retaining Meaning

Living in a Unicode World

Unicode Case Folding

Unicode Character Folding

Sorting and Collations

Case-Insensitive Sorting

Differences Between Languages

Unicode Collation Algorithm

Unicode Sorting

Specifying a Language

Customizing Collations

Reducing Words to Their Root Form

Algorithmic Stemmers

Using an Algorithmic Stemmer

Dictionary Stemmers

Hunspell Stemmer

Installing a Dictionary

Per-Language Settings

Creating a Hunspell Token Filter

Hunspell Dictionary Format

Choosing a Stemmer

Stemmer Performance

Stemmer Quality

Stemmer Degree

Making a Choice

Controlling Stemming

Preventing Stemming

Customizing Stemming

Stemming in situ

Is Stemming in situ a Good Idea

Stopwords: Performance Versus Precision

Pros and Cons of Stopwords

Using Stopwords

Stopwords and the Standard Analyzer

Maintaining Positions

Specifying Stopwords

Using the stop Token Filter

Updating Stopwords

Stopwords and Performance

and Operator

minimum\_should\_match

Divide and Conquer

Controlling Precision

Only High-Frequency Terms

More Control with Common Terms

Stopwords and Phrase Queries

Positions Data

Index Options

Stopwords

common\_grams Token Filter

At Index Time

Unigram Queries

Bigram Phrase Queries

Two-Word Phrases

Stopwords and Relevance

Synonyms

Using Synonyms

Formatting Synonyms

Expand or contract

Simple Expansion

Simple Contraction

Genre Expansion

Synonyms and The Analysis Chain

Case-Sensitive Synonyms

Multiword Synonyms and Phrase Queries

Use Simple Contraction for Phrase Queries

Synonyms and the query\_string Query

Symbol Synonyms

Typoes and Mispelings

Fuzziness

Fuzzy Query

Improving Performance

Fuzzy match Query

Scoring Fuzziness

Phonetic Matching

Aggregations

High-Level Concepts

Buckets

Metrics

Combining the Two

Aggregation Test-Drive

Adding a Metric to the Mix

Buckets Inside Buckets

One Final Modification

Building Bar Chart

Looking at Time

Returning Empty Buckets

Extended Example

The Sky’s the Limit

Scoping Aggregations

Filtering Queries and Aggregations

Filtered Query

Filter Bucket

Post Filter

Recap

Sorting Multivalue Buckets

Intrinsic Sorts

Sorting by a Metric

Sorting Based on “Deep” Metrics

Approximate Aggregations

Finding Distinct Counts

Understanding the Trade-offs

Optimizing for Speed

Calculating Percentiles

Percentile Metric

Percentile Ranks

Understanding the Trade-offs

Significant Terms

significant\_terms Demo

Recommending Based on Popularity

Recommending Based on Statistics

Controlling Memory Use and Latency

Fielddata

Aggregations and Analysis

High-Cardinality Memory Implications

Limiting Memory Usage

Fielddata Size

Monitoring fielddata

Circuit Breaker

Fielddata Filtering

Doc Values

Enabling Doc Values

Preloading Fielddata

Eagerly Loading Fielddata

Global Ordinals

Index Warmers

Preventing Combinatorial Explosions

Depth-First Versus Breadth-First

Closing Thoughts

Geolocation

Geo-Points

Lat/Lon Formats

Filtering by Geo-Point

geo\_bounding\_box Filter

Optimizing Bounding Boxes

geo\_distance Filter

Faster Geo-Distance Calculations

geo\_distance\_range Filter

Caching geo-filters

Reducing Memory Usage

Sorting by Distance

Scoring by Distance

Geohashes

Mapping Geohashes

geohash\_cell Filter

Geo-aggregations

geo\_distance Aggregation

geohash\_grid Aggregation

geo\_bounds Aggregation

Geo-shapes

Mapping geo-shapes

precision

distance\_error\_pct

Indexing geo-shapes

Querying geo-shapes

Querying with Indexed Shapes

Geo-shape Filters and Caching

Modeling Your Data

Handling Relationships

Application-side Joins

Denormalizing Your Data

Field Collapsing

Denormalization and Concurrency

Renaming Files and Directories

Solving Concurrency Issues

Global Locking

Document Locking

Tree Locking

Nested Objects

Nested Object Mapping

Querying a Nested Object

Sorting by Nested Fields

Nested Aggregations

reverse\_nested Aggregation

When to Use Nested Objects

Parent-Child Relationship

Parent-Child Mapping

Indexing Parents and Children

Finding Parents by Their Children

min\_children and max\_children

Finding Children by Their Parents

Children Aggregation

Grandparents and Grandchildren

Practical Considerations

Memory Use

Global Ordinals and Latency

Multigenerations and Concluding Thoughts

Designing for Scale

The Unit of Scale

Shard Overallocation

Kagillion Shards

Capacity Planning

Replica Shards

Balancing Load with Replicas

Multiple Indices

Time-Based Data

Index per Time Frame

Index Templates

Retiring Data

Migrate Old Indices

Optimize Indices

Closing Old Indices

Archiving Old Indices

User-Based Data

Shared Index

Faking Index per User with Aliases

One Big User

Scale Is Not Infinite

Administration, Monitoring, and Deployment

Monitoring

Marvel for Monitoring

Cluster Health

Drilling Deeper: Finding Problematic Indices

Blocking for Status Changes

Monitoring Individual Nodes

indices Section

OS and Process Sections

JVM Section

Threadpool Section

FS and Network Sections

Circuit Breaker

Cluster Stats

Index Stats

Pending Tasks

cat API

Production Deployment

Hardware

Memory

CPUs

Disks

Network

General Considerations

Java Virtual Machine

Transport Client Versus Node Client

Configuration Management

Important Configuration Changes

Assign Names

Paths

Minimum Master Nodes

Recovery Settings

Prefer Unicast over Multicast

Don’t Touch These Settings!

Garbage Collector

Threadpools

Heap: Sizing and Swapping

Give Half Your Memory to Lucene

Don’t Cross 32 GB!

Swapping Is the Death of Performance

File Descriptors and MMap

Revisit This List Before Production

Post-Deployment

Changing Settings Dynamically

Logging

Slowlog

Indexing Performance Tips

Test Performance Scientifically

Using and Sizing Bulk Requests

Storage

Segments and Merging

Other

Rolling Restarts

Backing Up Your Cluster

Creating the Repository

Snapshotting All Open Indices

Snapshotting Particular Indices

Listing Information About Snapshots

Deleting Snapshots

Monitoring Snapshot Progress

Canceling a Snapshot

Restoring from a Snapshot

Monitoring Restore Operations

Canceling a Restore

Additional Topics

1. How to monitor an Elasticsearch cluster and to understand what the monitoring analytics means, when to take action and what to look for and where + typical answers to common issues. ( Elasticsearch)

2.       Troubleshooting connectivity between I3 and Elasticsearch - when to worry and what actions to take, and when to ignore. ( I3 )

3.       How to troubleshoot bad performance in an Elasticsearch node to find possible bottlenecks - and what can be done to improve performance. ( Elasticsearch )

4.       How to use Sense to query Elasticsearch directly - training as well as words of warning. ( Elasticsearch )

5.       Procedures for backup and restore as well as running the system with some nodes offline, e.g. in the case a server needs to be replaced or some servers (Linux) need to be patched. (Elasticsearch )

6.       Securing a cluster, what options are there to limit access to Elasticsearch with and without commercial plugins like X-pack. Focus on best practice, i.e. checking that the cluster is secured behind firewalls and passwords. ( Elasticsearch )

7.       How to configure Beats and Logstash for logging into Elasticsearch for visualization in Kibana. ( Elasticsearch, I3 )

8.       Session about Elasticsearch configuration - working with analyzers and understandinng Elasticsearch mapping (configuration). ( Elasticsearch )

9.       Session about I3 search service configuration for new use cases - walkthrough of how to configure searchers and completers for betting understanding by the BUGS team. ( I3 )

10.   Planning for the future - caching, scaling up the cluster, Java settings etc. ( Elasticsearch )