Cassandra Performance Tuning

Apache Cassandra:
Performance Tuning and Cassandra

Learning objectives

- **Discuss general performance tuning**
- Discuss common performance anti-patterns
- Identify performance tuning goals
- Introduce class Cassandra use case
- Analyze class data model
- Identify performance tuning methodologies

Question

DATASTAX

When do we need to tune performance?

When do we need to tune performance?

DATASTAX

- Optimizing
  - When things work but could be better
- Troubleshooting
  - Fixing a problem that impedes performance
  - Could actually be broken
  - Could just be slow
    - In clusters, something broken can manifest as slow performance

Question

DATASTAX

What are some examples of performance related complaints an admin might receive regarding Cassandra?

Performance-related complaints

- "It's slow"
- "Certain queries are slow"
- "Program X that uses the cluster is slow"
- "A node went down"

Learning objectives

- Discuss general performance tuning
- **Discuss common performance anti-patterns**
- Identify performance tuning goals
- Introduce class Cassandra use case
- Analyze class data model
- Identify a performance tuning methodologies

Question

DATASTAX

How *not* to approach performance-related problems?

How *not* to approach solving perfomance-related problems?

- Streetlight Anti-Method
- Random Change Anti-Method
- Blame Someone Else Anti-Method

Streetlight Anti-Method

DATASTAX

- Absence of deliberate methodology
- User analyzes performance by:
  - choosing tools that are familiar
  - using tools found on internet
  - random testing to see if anything obvious shows up
- Tuning performance is then done by trial-and-error
- Tune things one at a time to see if something changes

Random Change Anti-Method

DATASTAX

- User guesses as to where problem might be.
- Changes things and observes performance
- Verifies result of each change by studying a metric such as latency or throughput
- When changes aren't understood, worse problems can occur during peak production load

Blame Someone Else Anti-Method

DATASTAX

- Assume problem lies with another system or component
- Hypothesize what the issue may be
- Redirect issue to the team responsible
- Repeat as necessary until a problem is actually found

Question

DATASTAX

In performance tuning, what are we trying to improve?

What are we trying to improve?

DATASTAX

- Latency—How long a cluster, node, server or I/O subsystem takes to respond to a request.
- Throughput—How many transactions of a given size (or range) a cluster, node or I/O subsystem can complete in a given timeframe.

Question

DATASTAX

How are latency & throughput related?

How are latency & throughput related?

- Theoretically, they are independent of each other.
- However, change in latency can have a proportional effect on throughput.

Question

DATASTAX

Relative to the innate capabilities of a given resource at rest, what causes a change in latency and throughput?

Understanding Performance Tuning

DATASTAX

- Utilization
- Saturation
- Errors
- Availability

Question

DATASTAX

What is utilization? Saturation? Errors? Availability?

What is Utilization? Saturation? Errors? Availability?

- Utilization—how heavily are the resources being stressed.
- Saturation—the degree to which a resource has queued work it cannot service.
- Errors—recoverable failure or exception events in the course normal operation.
- Availability—whether a given resource can service work or not.

DATASTAX

Exercise 1: Load Working Data

Learning objectives

- Discuss general performance tuning
- Discuss common performance anti-patterns
- **Identify performance tuning goals**
- Introduce class Cassandra use case
- Analyze class data model
- Identify a performance tuning methodologies

Question

DATASTAX

What is the first step in achieving any performance tuning goal?

DATASTAX

What is the first step in achieving any kind of goal?

Setting a goal!

Question

DATASTAX

What are some examples of commonly heard Cassandra performance tuning goals?

Examples of common Cassandra goals

DATASTAX

- Reads should be *faster.*
- Writes to table X should be *faster.*
- The cluster should be able to complete X transactions per second.

Question

DATASTAX

What should a clearly defined performance goal take into account?

Attributes of clearly defined performance goal

DATASTAX

- Type of operation or query
  - Read or Write
  - SELECT, INSERT / UPDATE / DELETE
- Latency
  - Expressed as percentile rank. E.g. "95th percentile read latency is 2 ms"
- Throughput
  - Expressed as operations per second
- Size
  - Expressed in average bytes
- Duration
  - Expressed in minutes or hours
- Scope
  - Keyspace, table, query
- Example: "The cluster should be able to sustain 20,000 2KB read operations per second from table X for two hours with a 95th percentile read latency of 3ms."

Question

DATASTAX

Can we set *general* performance goals, like "faster reads"?

Can we set general goals like "faster reads"?

DATASTAX

- NO.
- Caching can improve reads, but which data do you cache?
- How do we make sure the right data is cached and not evicted?
- Does your data model support caching the right data?
- The only "silver bullet" for Cassandra performance is "keep everything in RAM"
  - Everything else requires deliberate, targeted tuning

Question

DATASTAX

How can achievement of a goal be verified?

How can achievement of a goal be verified?

- Timing hooks in your application
- Query tracing
- jmeter test plan
- Customizable cassandra-stress

Time in Computer Performance Tuning

Question

DATASTAX

How long is a millisecond?
Why do we care about milliseconds?

Why do we care about milliseconds?

DATASTAX

Event	Latency	Scaled
1 CPU cycle	0.3 ns	1 s
Level 1 cache access	0.9 ns	3 s
Level 2 cache access	2.8 ns	9 s
Level 3 cache access	12.9 ns	43 s
Main memory access (DRAM, from CPU)	120 ns	6 min
Solid-state disk I/O (flash memory)	50–150 µs	2–6 days
Rotational disk I/O	1–10 ms	1–12 months
Internet: San Francisco to New York	40 ms	4 years
Internet: San Francisco to United Kingdom	81 ms	8 years
Internet: San Francisco to Australia	183 ms	19 years
TCP packet retransmit	1–3 s	105–317 years
OS virtualization system reboot	4 s	423 years
SCSI command time-out	30 s	3 millennia
Hardware (HW) virtualization system reboot	40 s	4 millennia
Physical system reboot	5 m	32 millennia

Common Latency Timings in Cassandra

DATASTAX

- Reads from main memory should take between 36 and 130 microseconds
- Reads from an SSD should take between 100 microseconds and 12 milliseconds
- Reads from a SAS (Serial Attached SCSI) drive should take between 8 milliseconds and 40 milliseconds
- Reads from a SATA drive take more than 15 milliseconds

Learning objectives

DATASTAX

- Discuss general performance tuning
- Discuss common performance anti-patterns
- Identify performance tuning goals
- **Introduce class Cassandra use case**
- Analyze class data model
- Identify a performance tuning methodologies

Classroom use case and Cassandra story

DATASTAX

- Middle sized financial firm
- Uses Cassandra to manage distributed data
- 42 million stock quotes
- Driven by a particular set of queries

What queries drove this data model?

DATASTAX

- Retrieve information for a specific stock trade by trade ID
- Find all information about stock trades for a specific stock ticker and range timestamps.
- Find all information about stock trades that occurred on a specific date over a short period of time.

How do you characterize your workload?

How do you characterize your workload?

- What is the load being placed on your cluster?
  - Calling application or API
  - Remote IP address
- Who is causing the load?
  - Code path or stack trace
- Why is the load being called?
- What are the load characteristics?
  - Throughput
  - Direction (read/write)
  - Include variance (standard deviation) where appropriate
  - Keyspace and column family

How do you characterize your workload? (cont.)

DATASTAX

- How is the load changing over time and is there a daily pattern?
- Is your workload read heavy or write heavy?
- How big is your data (total bytes in cluster)?
- How much data on each node (bytes on node=data density)?
- Does active data fit in buffer cache?

Learning objectives

DATASTAX

- Discuss general performance tuning
- Discuss common performance anti-patterns
- Identify performance tuning goals
- Introduce class Cassandra use case
- **Analyze class data model**
- Identify performance tuning methodologies

Question

DATASTAX

How does the data model affect performance?

Impact of data model on performance

- Poorly shaped rows (too narrow or too wide)
- Hotspots (particular areas with a lot of reads/writes)
- Poor primary or secondary indexes
- Too many tombstones

Data model considerations

DATASTAX

- Understand how primary key affects performance
- Take a look at query patterns and adjust how tables are modeled
- See how replication factor and/or consistency level impact performance
- Change in compaction strategy can have a positive (or negative) impact
- Parallelize reads/writes if necessary
- Look at how moving infrequently accessed data can improve performance
- See how per column family cache is having an impact

Question

DATASTAX

What is the relationship between the data model and Cassandra's read path optimizations (key/row cache, bloom filters, index)?

Relationship between data model and Cassandra's read path optimizations?

- Nesting data
  - Nesting data allows for greater degree of flexibility in the column family structure.
- Model to keep most active data sets in cache
  - Frequently accessed data which are in cache can improve performance.

Exercise 2: Understand your Data Model

Learning objectives

- Discuss general performance tuning
- Discuss common performance anti-patterns
- Identify performance tuning goals
- Introduce class Cassandra use case
- Analyze class data model
- **Identify performance tuning methodologies**

Performance Tuning Methodology

DATASTAX

- Active Performance Tuning—Suspect there's a problem
  - Isolate problem(s) using tools provided
  - Determine if problem is in Cassandra, environment or both
  - Verify problems and test for reproducibility
  - Fix problems using tuning strategies provided
  - Test, test, and test again
    - Verify that your "fixes" did not introduce additional problems
- Passive Performance Tuning—Regular system "sanity checks"
  - Regularly monitor key health areas in Cassandra/Environment using tools provided
  - Identify and tune for future growth/scalability
  - Apply tuning strategies as needed

The USE Method as a tool for troubleshooting

DATASTAX

- Strategy devised by Brendan Gregg
  - www.brendangregg.com/USEmethod/use-linux.html
- Performs a health check of various system components to identify bottlenecks and errors
- Separated by component, type and metric to narrow scope and find location of problem

Exercise 3: Use jmeter to get baseline

Demo 1: Use jmeter to collect metrics

Summary

- Reasons to tune performance are optimization and troubleshooting
- Anti-methods are common but not ideal ways to tune performance
- Performance tuning is used to lower latency and increase throughput
- The USE Method is a tool provided to narrow down performance tuning problem areas

Review Questions

- What are the two things performance tuning improves?
- What are two types of performance tuning methodologies?
- What tool can be used to get a performance baseline?