Similarity Ranking-based Root Cause Analysis for Distributed Systems

Executive Summary:

In this report, the root cause analysis results obtained from a similarity ranking algorithm on the data provided by a large enterprise tech company is discussed. The key hypothesis that will be tested is that a similarity ranking can be computed across metrics generated at different hierarchical layers in a distributed system and that ranking can be used for complex root cause analysis for a distributed system.

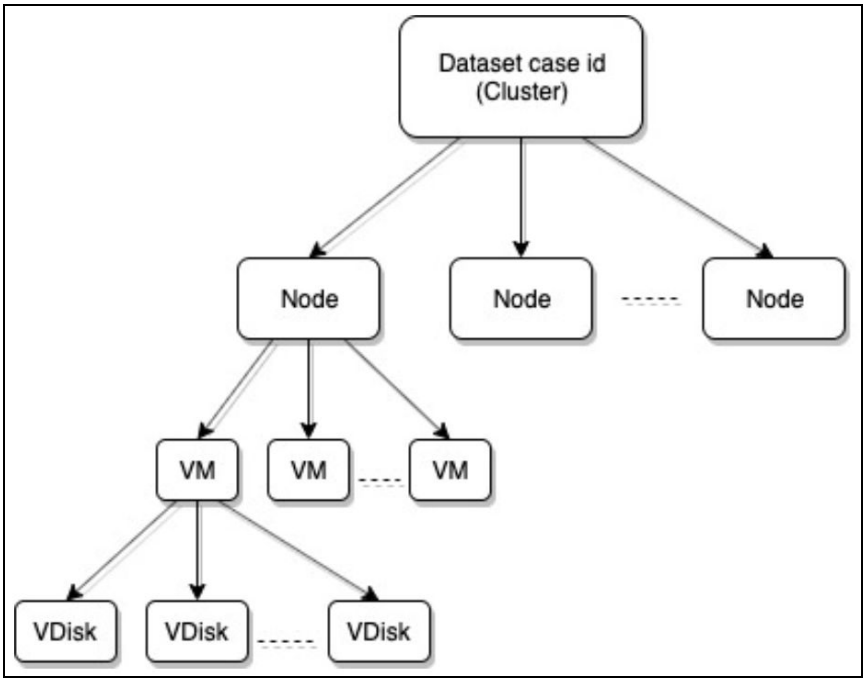
Table of Content

Key Terminologies:

- Issue: Fault metric for which root cause analysis is being conducted.
- Candidates: The metric which is being related to the fault metric.

Data:

A corpus of metrics data was provided from a distributed system, which is organized in a (cluster (dataset caseID) -> node -> vm -> vdisk) hierarchy.



There are about 50 unique caseIDs. Each caseID (cluster) noted to have 3-5 different nodes. Each node noted to have 3-5 different VMs. Each VM noted to have 20-30 vdisks. Overall, there are about ~9672 different metric measurement files. Each data collection has about 39 different metics:

1. controller_avg_read_io_latency_usecs
2. controller_avg_read_io_size_kbytes
3. controller_avg_write_io_latency_usecs
4. controller_avg_write_io_size_kbytes
5. controller_frontend_ops

6. controller_frontend_read_ops
7. controller_frontend_write_ops
8. controller_misaligned_offset_reads
9. controller_misaligned_offset_writes
10. controller_num_read_iops
11. controller_num_write_iops
12. controller_oplog_drain_dest_hdd_bytes
13. controller_oplog_drain_dest_ssd_bytes
14. controller_random_ops
15. controller_random_read_bytes
16. controller_random_read_ops
17. controller_random_write_bytes
18. controller_random_write_ops
19. controller_read_io_bandwidth_kbps
20. controller_read_source_cache_dram_bytes
21. controller_read_source_estore_hdd_local_bytes
22. controller_read_source_estore_hdd_remote_bytes
23. controller_read_source_estore_ssd_local_bytes
24. controller_read_source_estore_ssd_remote_bytes
25. controller_read_source_extent_cache_bytes
26. controller_read_source_oplog_bytes
27. controller_seq_read_ops
28. controller_seq_write_ops
29. controller_storage_tier_ssd_pinned_usage_bytes
30. controller_storage_tier_ssd_usage_bytes
31. controller_vdisk_cpu_time_usecs
32. controller_write_dest_estore_hdd_bytes
33. controller_write_dest_estore_ssd_bytes
34. controller_write_dest_oplog_bytes
35. controller_write_io_bandwidth_kbps
36. controller_wss_120s_read_mb
37. controller_wss_120s_write_mb
38. hypervisor_num_read_iops

39. hypervisor_num_write_iops

The metrics are sampled at a 2 min interval with a window of size 30 min-2 hr. However, there are a few empty files.

Methodology:

A time-series similarity-based approach has been adopted where we presume two time series with higher distance score will have low correlations. The correlation computation we use is considerably different from traditional (Pearson, Kendall, Spearman) approaches. The traditional point-to-point approach does not work for the root cause analysis in distributed systems for the following reasons, as follows:

- *Phase shift*: The time series distribution alters significantly in the event of an incident. The effect of incident propagates at different speeds at different hierarchy. It in turn causes a phase shift for different metrics meaning not all metrics are starting from the same time instant. This is mostly relevant for the cross-hierarchy correlation determination.

- *Difference in order-of-magnitudes*: The metrics at different layers tend to have different order of magnitudes. Traditional point-to-point correlation computation fails to work in these scenarios due their linear dependencies with the metric values.

- *Computation time*: The traditional point-wise computation use quadratic-time algorithm, which does not scale well.

Our similarity-based approach resolves these constraints. As shown in the following chart, a correlation is being drawn between controller_avg_write_io_latency_usecs (issue metric) of the order of 10^4, lasting from 5:02 AM at 09/05/2019 to 5:50 AM at 09/05/2019, and controller_num_write_iops_vdisk_db32c386-f0be-49d3-a0b4-c6e07b4485e7 (candidate metric) of the order of 10^1, lasting from 5:00 AM at 09/05/2019 to 5:58 AM at 09/05/2019. All times are in PST.

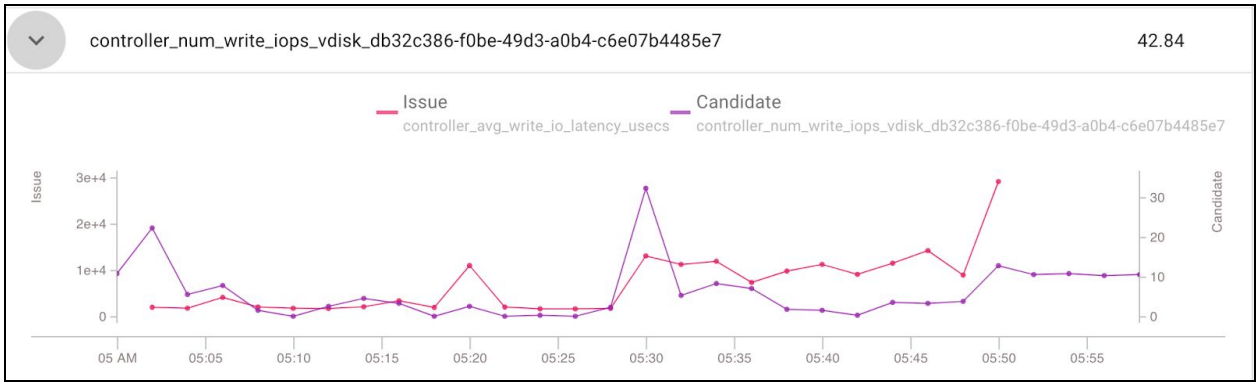
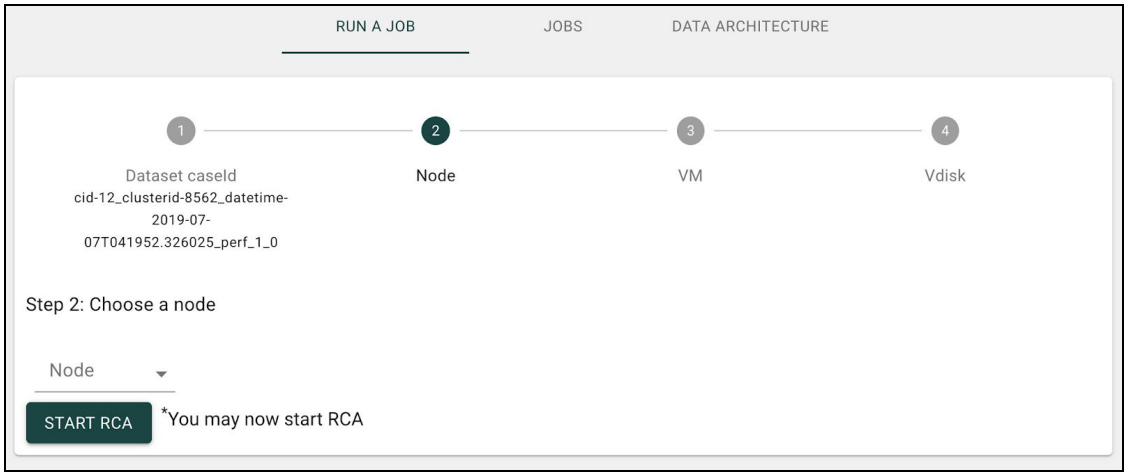


Figure 1: Correlation distance computation between controller_avg_write_io_latency_usecs (issue metric) and controller_num_write_iops_vdisk_db32c386-f0be-49d3-a0b4-c6e07b4485e7 (candidate metric)

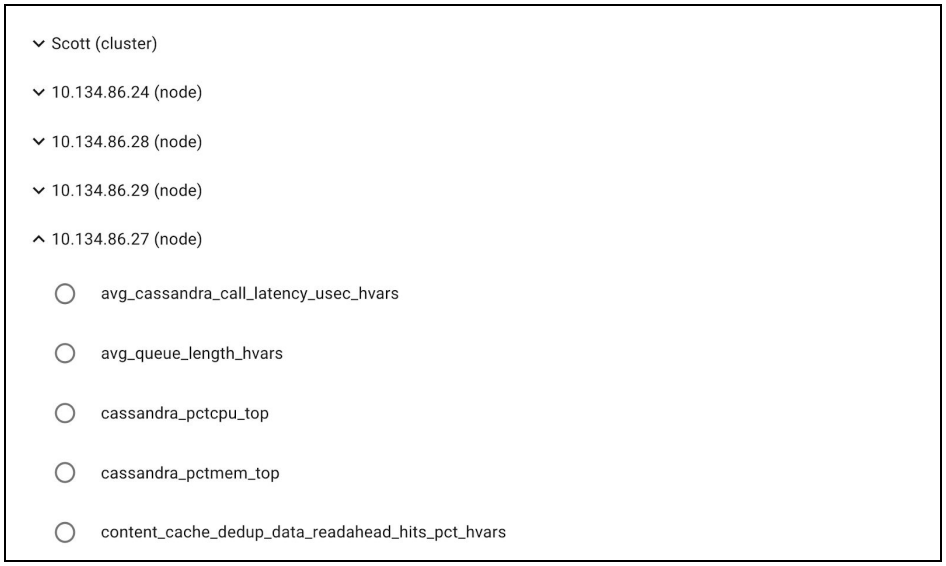
User Sequence:

The software app provides a step-by-step flow for the similarity ranking-based root cause analysis.

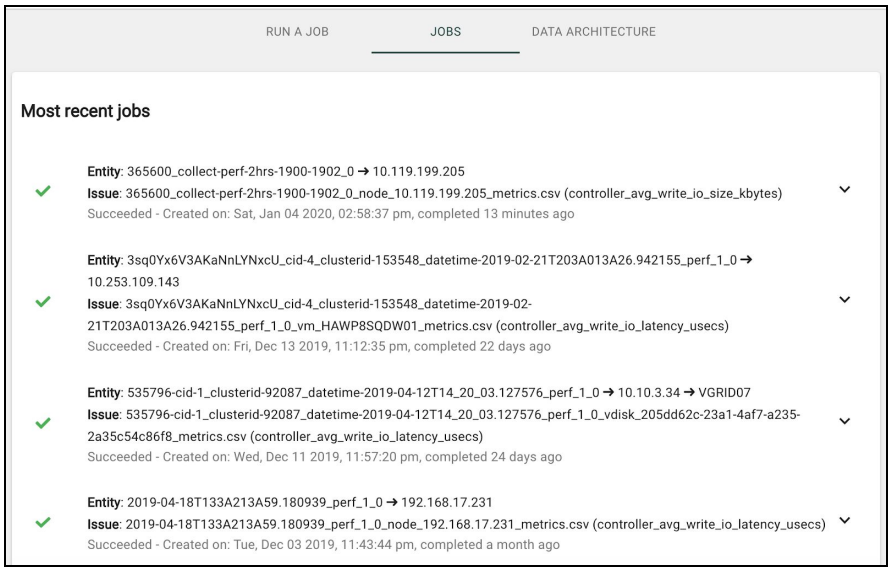


Step 1: Choose the appropriate subtree for the similarity ranking-based root cause analysis.

Step 2: Choose the appropriate issue metric from the subtree. (Once a subtree is created, the root cause analysis will be limited to that subtree; meaning for a given issue metric all the metrics within the subtree will be taken as the candidate metric.)



Step 3: Go to the Job tab, wait for 1-3 min for the job to be completed. The computation time depends upon the subtree size. All the jobs are organized in a reverse-chronological order.



Step 4: Each table entry has following fields.

1. Entity: This field specifies the chosen substree. For example, for the following job the entity is (3sq0Yx6V3AKaNnLYNxcU_cid-4_clusterid-153548_datetime-2019-02-21T203A013A26.942155_perf_1_0 → 10.253.109.143). It means the chosen subtree can specified by cluster or datasetID:

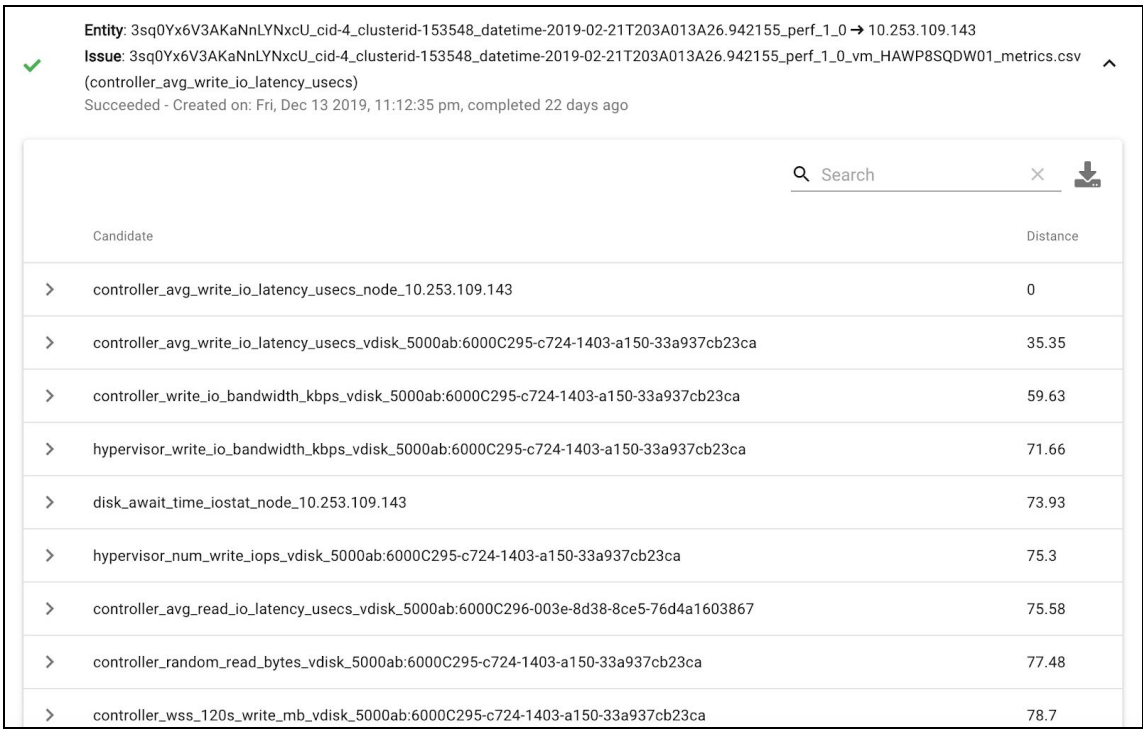
3sq0Yx6V3AKaNnLYNxcU_cid-4_clusterid-153548_datetime-2019-02-21T203A013A26.942155_perf_1_0 and node: 10.253.109.143

2. Issue: This field specifies the issue metric. For example, for the following job the issue metric is controller_avg_write_io_latency_usecs from the file 3sq0Yx6V3AKaNnLYNxcU_cid-4_clusterid-153548_datetime-2019-02-21T203A013A26.942155_perf_1_0_vm_HAWP8SQDW01_metrics.csv.

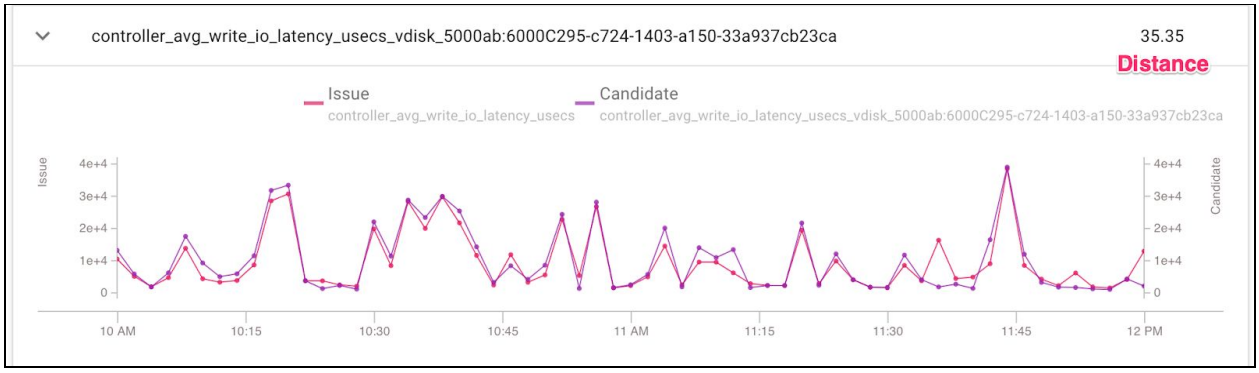
3. Status: The Status field shows the completion status of the job and when the job was completed.

Entity: 3sq0Yx6V3AKaNnLYNxcU_cid-4_clusterid-153548_datetime-2019-02-21T203A013A26.942155_perf_1_0 ➔ 10.253.109.143
Issue: 3sq0Yx6V3AKaNnLYNxcU_cid-4_clusterid-153548_datetime-2019-02-21T203A013A26.942155_perf_1_0_vm_HAWP8SQDW01_metrics.csv (controller_avg_write_io_latency_usecs)
Succeeded - Created on: Fri, Dec 13 2019, 11:12:35 pm, completed 22 days ago

Step 5: The result consists of a ranked list of candidates as possible causal factors for the issue.



Step 6: The correlation between a particular candidate against the issue can be further visually analyzed by clicking on the accordion. The distance is used as the similarity measure for the ranking.



Fault Tolerance:

The software app is designed to handle different failure conditions gracefully. These conditions are:

(a) Either the issue metric or the candidate metric don't have any data point.
(b) There is no overlap between issue metric and candidate metric.

A Few Representative Examples: