

EXAMTOPICS

- Expert Verified, Online, Free.

≡ MENU



🔍 Google Discussions



Exam Professional Machine Learning Engineer All Questions

View all questions & answers for the Professional Machine Learning Engineer exam

Go to Exam

EXAM PROFESSIONAL MACHINE LEARNING ENGINEER TOPIC 1 QUESTION 32 DISCUSSIO..

Actual exam question from Google's Professional Machine Learning Engineer

Question #: 32

Topic #: 1

[\[All Professional Machine Learning Engineer Questions\]](#)

You developed an ML model with AI Platform, and you want to move it to production. You serve a few thousand queries per second and are experiencing latency issues. Incoming requests are served by a load balancer that distributes them across multiple KubeFlow CPU-only pods running on Google Kubernetes Engine (GKE). Your goal is to improve the serving latency without changing the underlying infrastructure. What should you do?

- A. Significantly increase the max_batch_size TensorFlow Serving parameter.
- B. Switch to the tensorflow-model-server-universal version of TensorFlow Serving.
- C. Significantly increase the max_enqueued_batches TensorFlow Serving parameter.
- D. Recompile TensorFlow Serving using the source to support CPU-specific optimizations. Instruct GKE to choose an appropriate baseline minimum CPU platform for serving nodes.


Show Suggested Answer

by  DucLee3110 at July 1, 2021, 7:58 a.m.

Comments

Type your comment...

Submit

  **Y2Data** Highly Voted  3 years, 1 month ago

D is correct since this question is focusing on server performance which development env is higher than production env. It's already throttling so increase the pressure on them won't help. Both A and C is essentially doing this. B is a bit mysterious, but we definitely know that D would work.

   upvoted 27 times

  **mousseUwU** 3 years ago

I think it's D too

   upvoted 2 times

  **desertlotus1211** Most Recent  22 hours ago

max_batch_size: Increasing the max_batch_size parameter allows TensorFlow Serving to process more requests in a single batch. This can improve throughput and reduce latency, especially in high-query environments, as it allows more efficient utilization of CPU resources by processing larger batches of requests at once.

Answer A

   upvoted 1 times

  **taksan** 2 months ago

Selected Answer: D

I think the correct is D, because the question is about reducing latency. As for A, increasing the batch size might event hurt latency if the system is overwhelmed to serve more multiple requests

   upvoted 1 times

  **chirag2506** 3 months, 4 weeks ago

Selected Answer: D

it is D


   upvoted 1 times

  **PhilipKoku** 4 months, 2 weeks ago

Selected Answer: C


C) Batch enqueued

   upvoted 1 times

  **pinimichele01** 6 months, 1 week ago

Selected Answer: D

increasing the max_batch_size TensorFlow Serving parameter, is not the best choice because increasing the batch size may not necessarily improve latency. In fact, it may even lead to higher latency for individual requests, as they will have to wait for the batch to be filled before processing. This may be useful when optimizing for throughput, but not for serving latency, which is the primary goal in this scenario.

   upvoted 1 times

  **pico** 11 months, 2 weeks ago

Selected Answer: C

https://github.com/tensorflow/serving/blob/master/tensorflow_serving/batching/README.md#batch-scheduling-parameters-and-tuning

A may help to some extent, but it primarily affects how many requests are processed in a single batch. It might not directly address latency issues.

D is a valid approach for optimizing TensorFlow Serving for CPU-specific optimizations, but it's a more involved process and might not be the quickest way to address latency issues.

   upvoted 4 times

  **ichbinnoah** 11 months, 2 weeks ago

Selected Answer: A

I think A is correct, as D implies changes to the infrastructure (question says you must not do that).

   upvoted 1 times

  **edoo** 7 months, 3 weeks ago

This is purely a software optimization and on how GKE handles requests. GKE should be able to choose different CPU types for nodes within the same cluster, which doesn't represent a change in architecture.

   upvoted 1 times

  **tavva_prudhvi** 1 year, 2 months ago

Selected Answer: D

increasing the max_batch_size TensorFlow Serving parameter, is not the best choice because increasing the batch size may

not necessarily improve latency. In fact, it may even lead to higher latency for individual requests, as they will have to wait for the batch to be filled before processing. This may be useful when optimizing for throughput, but not for serving latency, which is the primary goal in this scenario.



   upvoted 1 times

  **harithacML** 1 year, 3 months ago

Selected Answer: D

max_batch_size parameter controls the maximum number of requests that can be batched together by TensorFlow Serving. Increasing this parameter can help reduce the number of round trips between the client and server, which can improve serving latency. However, increasing the batch size too much can lead to higher memory usage and longer processing times for each batch.

   upvoted 1 times

  **Liting** 1 year, 3 months ago

Selected Answer: D

Definetely D
to improve the serving latency of an ML model on AI Platform, you can recompile TensorFlow Serving using the source to support CPU-specific optimizations and instruct GKE to choose an appropriate baseline minimum CPU platform for serving nodes, this way GKE will schedule the pods on nodes with at least that CPU platform.

   upvoted 1 times

  **M25** 1 year, 5 months ago

Selected Answer: D

Went with D

   upvoted 1 times

  **SergioRubiano** 1 year, 7 months ago

Selected Answer: A

A is correct. max_batch_size TensorFlow Serving parameter

   upvoted 2 times

  **Yajnas_arpohc** 1 year, 7 months ago

Selected Answer: A

CPU-only: One Approach

If your system is CPU-only (no GPU), then consider starting with the following values: num_batch_threads equal to the number of CPU cores; max_batch_size to a really high value; batch_timeout_micros to 0. Then experiment with batch_timeout_micros values in the 1-10 millisecond (1000-10000 microsecond) range, while keeping in mind that 0 may be the optimal value.

https://github.com/tensorflow/serving/tree/master/tensorflow_serving/batching

   upvoted 3 times

  **frangm23** 1 year, 6 months ago

In that very link, what it says is that max_batch_size is the parameter that governs the latency/troughput tradeoff, and as I understand, the higher the batch size, the higher the throughput, but that doesn't assure that latency will be lower.

I would go with D

   upvoted 4 times

  **Omi_04040** 1 year, 10 months ago

Answer: D

<https://www.youtube.com/watch?v=fnZTVQ1SnDg>

   upvoted 1 times

  **wish0035** 1 year, 10 months ago

Selected Answer: D

ans: D

   upvoted 1 times

  **sachinxshrivastav** 2 years, 2 months ago

Selected Answer: D

D is the right one

   upvoted 1 times

[Load full discussion...](#)

Start Learning for free



Social Media

[Facebook](#) , [Twitter](#)

[YouTube](#) , [Reddit](#)

[Pinterest](#)



We are the biggest and most updated IT certification exam material website.

Using our own resources, we strive to strengthen the IT professionals community for free.



© 2024 ExamTopics

ExamTopics doesn't offer Real Microsoft Exam Questions. ExamTopics doesn't offer Real Amazon Exam Questions. ExamTopics Materials do not contain actual questions and answers from Cisco's Certification Exams.

CFA Institute does not endorse, promote or warrant the accuracy or quality of ExamTopics. CFA® and Chartered Financial Analyst® are registered trademarks owned by CFA Institute.