Exploring Serverless Computing for Neural Network Training

IEEE 2018

Authors: Lang Feng, Prabhakar Kudva, Dilma Da Silva, Jiang Hu
Texas A&M University & IBM Research

Group 4: James Haines
Peter Chu
Swarna Shenoy

1

Outline

1. Paper Overview
2. Related Work
3. Summary of new approach
4. Key contributions
5. Author's Evaluation
6. Author's Conclusion
7. Critique: Strengths
8. Critique: Weaknesses
9. Critique: Evaluation
10. Critique: Improvements

2

Paper Overview: The Big Picture/ Problem

Machine Learning - Where and how do we train?

- Mostly local or VM Based as of right now.
  - Can be unnecessarily costly
  - Time consuming
- Why go serverless?
  - Minimize the amount of idle cost
  - Distribute learning processes for speedup
- Current Status (as-of/prior-to this paper)
  - Lightweight machine learning algorithms trained serverless
  - Neural Networks? Trained offline - scored online

3

Paper Overview: Why we care

- Neural Networks vs Other ML Models:
  - For a large-and-growing number of tasks Neural Networks(NN) are state of the art
  - Training Models online allows for:
    - Larger volumes of training
    - Static storage needs
  - What does that mean?
    - Sophisticated online NN limited to large companies
    - Opens Machine-Learning-as-a-Service (MLaaS) to more people (devs and clients)

4

Related Work

Cost benefits of FaaS vs other runtimes:

- Awesome serverless Git. https://github.com/anaibol/awesome-serverless.

Broadening the use of serverless runtimes:

- E. Jonas, S. Venkataraman, I. Stoica, and B. Recht, "Occupy the cloud: Distributed computing for the 99%," Computing Research Repository, 2017. (**Pywren** to deploy)
- M. Malawski, K. Figiela, A. Gajek, and A. Zima, "Benchmarking heterogeneous cloud functions," in Euro-Par 2017: Parallel Processing Workshops (D. B. Heras and L. Bougé, eds.), pp. 415–426, 2018 (**HyperFlow** for performance evaluation)

5

Related Work (Contd.)

- M. Malawski, "Towards serverless execution of scientific workflows - hyperflow case study," in WORKS@SC, November 2016. (**HyperFlow**)

ML on Serverless Architectures:

- Google Cloud, "Building a Serverless ML Model." https://cloud.google.com/solutions/building-a-serverless-ml-model.

- V. Ishakian, V. Muthusamy, and A. Slominski, "Serving deep learning models in a serverless platform," Computing Research Repository, 2017. (**Latency** is investigated)

6

Summary of New Approach

Key Concept: Training Large Models (Models that cannot be trained within a single FaaS instance)

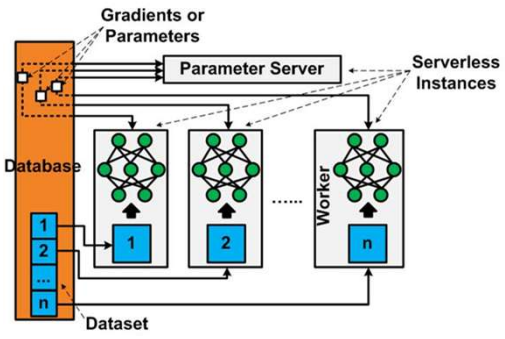
1. Data transfer and parallelism:
   a. Problem: At time of writing, no direct way to transfer data between serverless instances
   b. Solution: Databases store information between stages of training
   c. Implications: Data and models have additional costs…
      i. Transfer latencies between sources and destinations
      ii. Warm up latency
2. Data Parallelism for Neural Network Training:
3. Optimizing parallelism structure for Serverless Training
4. Cost and performance - ratio optimization

7

Summary of New Approach

Key Concept: Training Large Models (Models that cannot be trained within a single FaaS instance)

1. Data transfer and parallelism:
2. Data Parallelism for Neural Network Training:
   a. Problem: Limited time and space availability for FaaS
   b. Solution: Horizontal Scalability Saves the Day
      i. Divide and Conquer
      ii. How? ------>
3. Optimizing parallelism structure for Serverless Training
4. Cost and performance - ratio optimization

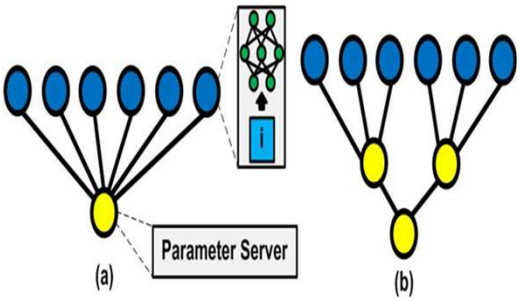


8

Summary of New Approach

Key Concept: Training Large Models (Models that cannot be trained within a single FaaS instance)

1. Data transfer and parallelism:
2. Data Parallelism for Neural Network Training:
3. Optimizing parallelism structure for Serverless Training:
    a. Problem: More instances = Higher cost (unnecessary waste)
    b. Solution: Find the golden ratio of workers to parameter servers
        i. Which takes more time? >>>>
        ii. There is math supporting this optimization…
        Not going to explain it (You're welcome)...
        Okay I'll explain it a little...
4. Cost and performance - ratio optimization



9

Summary of New Approach

Key Concept: Training Large Models (Models that cannot be trained within a single FaaS instance)

1. Data transfer and parallelism:
2. Data Parallelism for Neural Network Training:
3. Optimizing parallelism structure for Serverless Training:
4. Cost and performance - ratio optimization:
    a. Motivation: Lambda is cheap compared to other cloud computing
    b. Question: How do we make the most of this
    c. Answer: There is an optimization formula that leverages dynamic programming…
    Also not going to dive deep into it here.

$C(z) = p \cdot n \cdot z \cdot t(z)$, $p = 1.63 \times 10 -8$ \$/(M B · s), n = Number of Lambda instances,
z = memory size (No less than minimum required to run application),
$t(z)$ = latency as a function of memory size

10

Key Contributions

- Propose serverless computing structures for training large deep neural networks through data parallelism.
- Develope parallel structure to reducing training latency.
- Optimize cost to performance ratio for serverless neural network training.
- Show the benefits of serverless hyperparameter optimization for smaller models.
- Outlining novel serverless runtimes for further investigation to overcome the limitations for larger models.

11

Author's Evaluation

Experiment:

For Case A, the number of training iterations is 50. The training is done by 100 workers, each of which has 512MB memory.

For Case B, the number of training iterations is 20. The training is done by 100 workers, each of which has 1536MB memory.

Latency:

Optimization:

Latency between serverless instance and database depends

On location of the database relative to the instance,

traffic on networks, multi-tenancy

Structure

Structure	Latency for Case A (s)	Latency for Case B (s)
*[1,5,22,100]	569.55	789.20
[1,100]	1216.97	1848.05
[1,2,100]	878.78	1195.97
[1,5,100]	650.66	866.29
[1,25,100]	616.67	920.22
[1,2,10,100]	570.66	823.25
[1,5,50,100]	604.90	890.91
[1,10,50,100]	534.28	838.89
[1,2,10,50,100]	585.25	883.96
[1,5,20,50,100]	578.22	868.78

12

Author's Evaluation (Contd.)

Training accuracy and convergence rate:

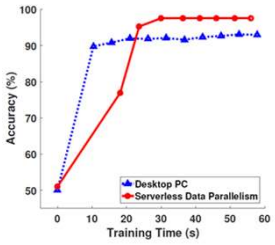


Figure 7. Training accuracy vs. training time for Case A.

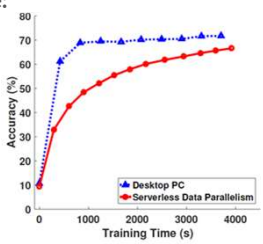


Figure 8. Training accuracy vs. training time for Case B.

In serverless computing, the parameter update is based on the average of gradients obtained from multiple workers. In the sequential training on desktop PC, by contrast, each parameter update is according to a single set of gradients from a single process.

13

Author's Evaluation(Contd.)

Result of Cost and Performance-Cost Ratio Optimization:

$C(z) = p \cdot n \cdot z \cdot t(z)$, $p = 1.63 \times 10\ {-8}$ \$/(M B · s), n = Number of Lambda instances, z = memory size (No less than minimum required to run application), $t(z)$ = latency as a function of memory size

$R(Z) = f/\ (C(z)\ t(z)\ )$, f = floating point operations for a computing task.

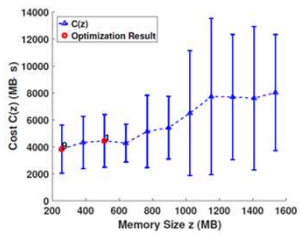


Figure 9. Cost per iteration under different Lambda instance memory sizes and optimization results.

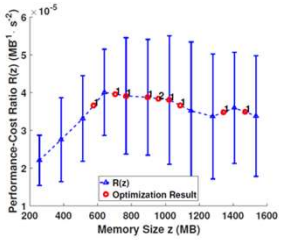


Figure 10. Performance-cost ratio per iteration under different Lambda instance memory sizes and optimization results.

14

Author's Evaluation(Contd.)

Results of hyperparameter tuning:

The hyperparameters can be decided either manually or through automated search such as random search, grid search and Bayesian optimization.

H ={h1, h2, ...} is a set of hyperparameters for a specific neural network explored are H= {H1,H2, ...}.

$$\sum_{i=1}^{|\mathcal{H}|} n_i \leq N.$$

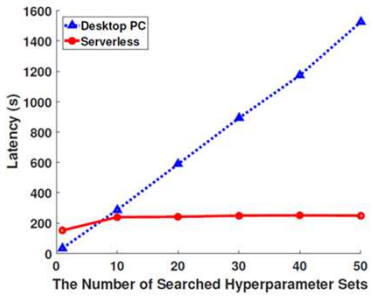


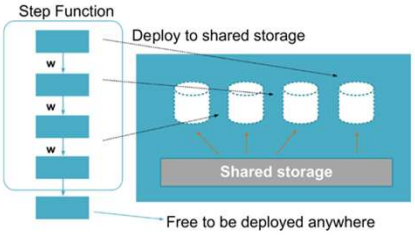
Figure 11. Computing latency versus the number of searched hyperparameter sets $|\mathcal{H}|$.

15

Author's Conclusions

For large models data parallelism were explored using various structures for composition of serverless instances.

The need to transfer data between subsequent serverless instances can result in latencies. So the author has proposed a potential design:



For smaller models, it was shown that serverless runtimes showed benefit for hyperparameter tuning.

16

Critique: Strengths

- Hyperparameter sets have a lot less latency in a serverless model than a traditional desktop
  - Parallelization has benefits that shine here cause of sequential execution is horizontally scaled
- Under certain circumstances, the serverless method yielded higher accuracy
- Training Times
  - The amount of time it takes to train a neural network based on a given data set is smaller when going serverless
  - Neural networks exercise the GPU because of the large number of operations that are not tightly coupled and thus can use the larger number of cores that GPUs come with
  - With a serverless model, there is `in theory` limitless horizontal scaling in order to maximize the cores offered by GPUs

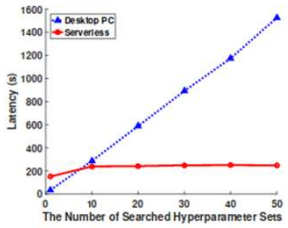
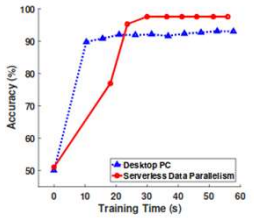


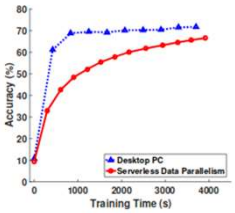
Figure 11. Computing latency versus the number of searched hyperparameter sets [9].



17

Critique: Weaknesses

- Accuracy
  - Under certain circumstances accuracy can be lower when done in a serverless method
  - This is a critical weakness because the goal of Machine Learning is to be as accurate as possible, a technology cannot be state of the art in machine learning if it isn't more accurate than other methods, or it doesn't offer something computationally novel such as security and privacy
- Startup Latency
  - Serverless runtimes are instantiated on infrastructure via resource scheduling by the service provider in a manner invisible to the end user
  - This would mean there would latency for each container that is spun up, which would not occur if neural networks are executed in the same machine
  - A possible mitigation is if containers with less overhead were implemented
- Data Transfer Latency
  - If the result of operations need to be stored in a database, there will be networking latency as the result of neural network operations have to travel from the container to the target database
- Resource Availability & Consistency
  - The availability of GPU cores to lambda containers may not be consistent to Lambda container
  - Lambda also has a limit of 15 minutes
  - Also the user cannot fully control the type of GPU available to the lambda container



18

Critique: Evaluation

- Should serveless machine learning be adopted?
  - It depends
- Increased Cost
  - Startup and data transfer latency could result in increased costs
  - The cheaper options ends up being based upon ratio optimization
- Decreased Cost
  - Reduced training times
  - Minimal idle time as operations don't wait for resources
- Optimized Cost will need to balance different cost factors
- Accuracy
  - There are certain circumstances where accuracy is lower and some cases where accuracy is higher
  - Depending on testing enviornment and goals, serverless may or may not offer benefits to accuracy

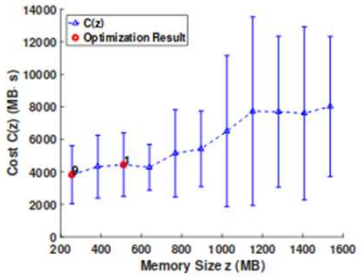


Figure 9. Cost per iteration under different Lambda instance memory sizes and optimization results.

19

Critique: Improvements

- Optimizations
  - Lightweight containers
    - Slim: https://www.usenix.org/conference/nsdi19/presentation/zhuo
    - https://www.usenix.org/sites/default/files/conference/protected-files/atc18_slides_thalheim.pdf
  - Faster RPCs
    - It is commonly believed that datacenter networking software must sacrifice generality to attain high performance
    - Introduces eRPC, a new general-purpose remote procedure call (RPC) library that offers performance comparable to specialized systems, while running on commodity CPUs in traditional datacenter networks based on either lossy Ethernet or lossless fabrics
    - https://www.usenix.org/conference/nsdi19/presentation/kalia
- NSDI (Networked Systems Design and Implementation) Conference
  - NSDI focuses on the design principles, implementation, and practical evaluation of networked and distributed systems
  - These conferences offer solutions to enhance to optimization point for doing serverless machine learning
- These improvements and yield a more attractive optimized costs

20

Gaps

- Does this research paper deliver?
  - Somewhat…
  - not all of the advances that are needed to make the switch from VM or a PC and Serverless training of neural networks.
  - Large discrepancy in accuracy which is what defines a ML models relevance
  - There still exist process changes that could further accelerate the functionality
  - However…
  - Insightful solutions for some problems
  - More than just a step in the right direction; several things went right.

21

Inspiring Quote:

"The best way to ensure that no one asks questions at the end of your presentation is to ask if there are any questions at the end of your presentation."

Thank you for listening. We hope you enjoyed our presentation.

22