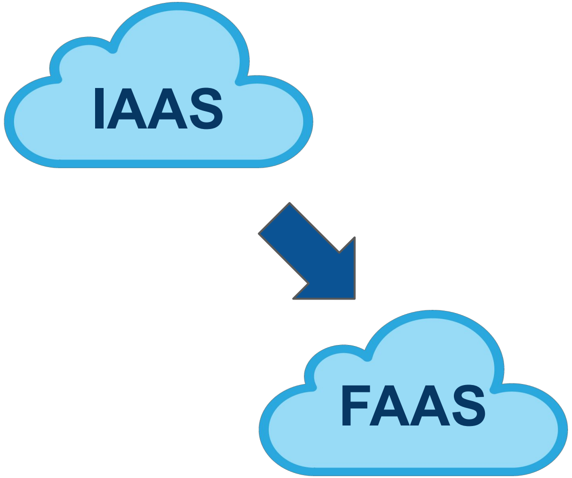
When Serverless Computing Meets Different Degrees of Customization for DNN Inference

Moohyun Song **Yoonseo Hur** **Kyungyong Lee**

Distributed Data Processing System Lab (DDPS Lab)
Department of Computer Science
Kookmin University, South Korea

Advancements in Cloud Computing



- Cloud computing
  - On-demand resources provisioning
  - Flexible pricing
- Infrastructure-as-a-Service (IaaS)
  - Aid to build highly-available system easily
- Function-as-a-Service (FaaS) and Serverless Computing
  - Low instance management overhead

Emerging Various Serverless Computing Runtimes and Workload

- Post-GPF serverless computing
  - Development of varied serverless execution environments.
    - SPF : AWS Sagemaker, SCS : GCP Cloud Run
- Limited serverless applications
  - Need for quantitative performance comparisons of DNN models.

What We Are Going To Cover In This Paper

Q1. How do SPF, GPF and SCS compare in performance across most DNN models?

Q2. How does API endpoint protocol impact inference time in end-to-end response?

Q3. Which is Better in SCS for Performance: More Instances or Increased CPU Cores?

Q4. What is the Impact of Cold Start and the Need for Further Research?

Experiment Setup and Workload

- ❏ Evaluation Framework:
  - ❏ GPF : AWS Lambda
  - ❏ SPF : AWS SageMaker Serverless Inference
  - ❏ SCS : GCP CloudRun

GPF
: AWS Lambda

SPF
: AWS SageMaker

SCS
: GCP Cloud Run

- ❏ Data Processing:

	Network	Data Type	Framework
GPF	REST	JSON	TensorFlow
SPF	REST, gRPC	JSON (Containing protobuf input data)	TFServing
SCS	gRPC	protobuf data	TensorFlow

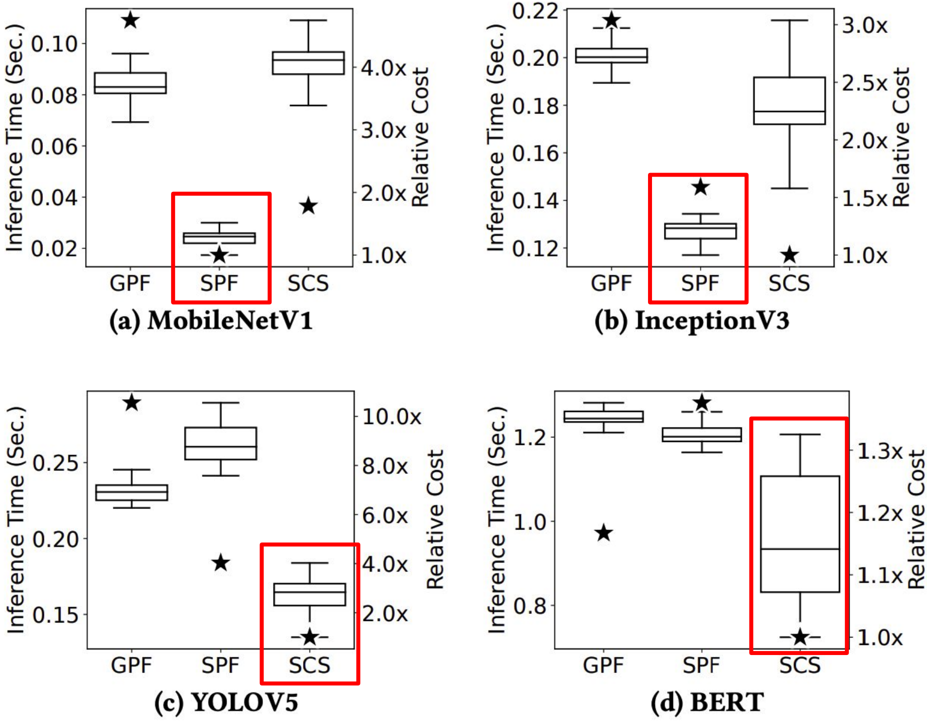
Experiment Setup and Workload

- Model and input/output dataset sizes (MB)

Model	GFLOPS	Model Size	Input Size		Output Size	
			gRPC	REST	gRPC	REST
MobileNetV1	1.15	18	0.574	3.014	0.0040	0.0268
InceptionV3	11.5	97	1.023	5.524	0.0040	0.0265
YOLOV5	16.5	28	4.688	24.547	8.172	63.5932
BERT	13.39	428	0.006	0.004	0.0001	0.0001

Q1. How do GPF, SPF and SCS compare in performance across DNN models?

SPF: Fastest Image Classification, SCS: Best for Large Model



- 40 simultaneous requests with 3GB memory
- SPF: Fastest in Image Classification Models
  - **TFServing** Boost Inference Time
- SCS: Fastest in YOLOv5, BERT
  - Skylake-SP Intel CPU Supporting **AVX512** in GCP Cloud Run

Impact of AVX512 on Inference Performance

	TFServing		Python	
	Disabled	Enabled	Disabled	Enabled
MobileNetV1	0.037	0.03	0.082	0.085
Inceptionv3	0.187	0.148	0.357	0.203

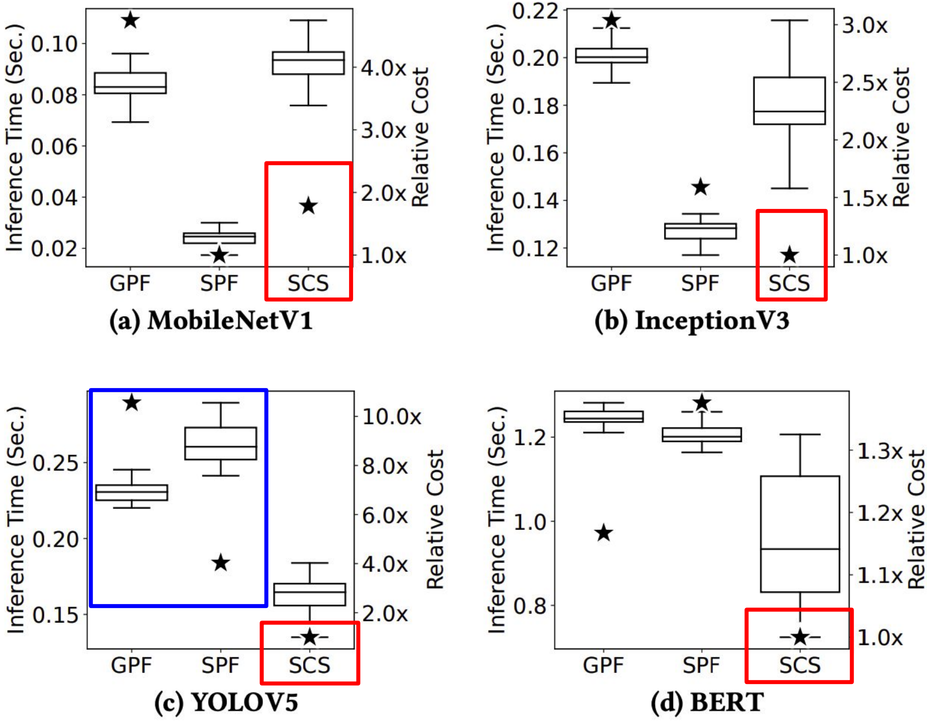
19.88% 19.73%

YOLOV5	0.345	0.276	0.331	0.255
BERT	1.827	1.246	1.830	1.294

25.90% 26.12%

- Larger Model Shows Greater Improvement in Image Classification
  - Small model has **higher overhead ratio** than large model.
- **Smaller models** perform better with **TFserving** in **GPF**
- **Large models**, **minimal** TFServing improvement Due to computation time

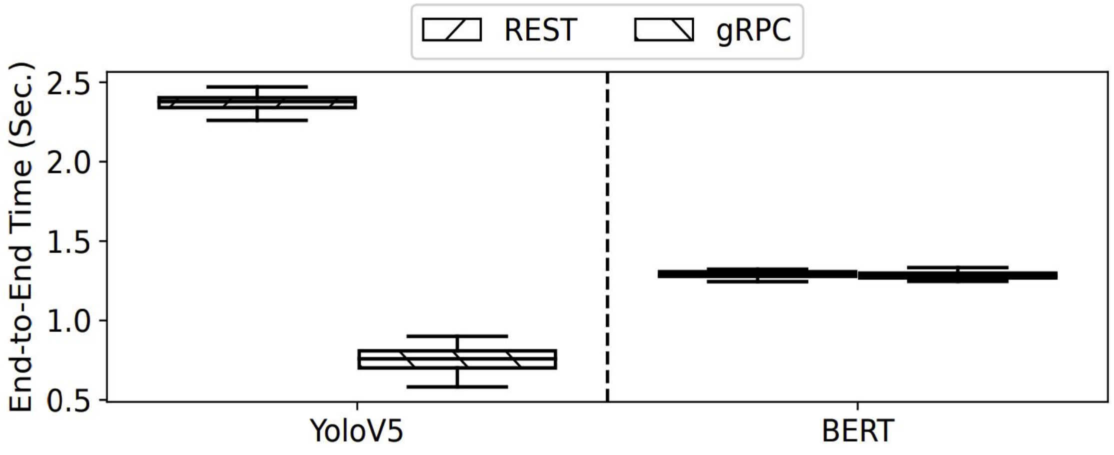
SCS: Most Cost-Efficient in Performance



- Cost calculation includes processing time
- SCS is the most cost-effective
  - higher degree of **scaling policy customization & lower cost**
- YOLOv5 inference faster with GPF, but cheaper with SPF
  - related to the **additional overhead**

Q2. How does API endpoint protocol impact inference time in end-to-end response?

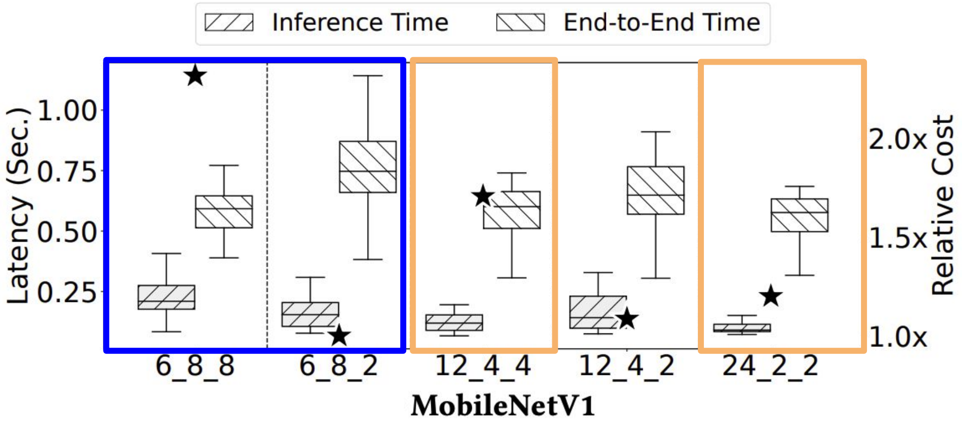
Large Data: gRPC vs REST Response Time



- ❑ Compare end-to-end time including network latency
- ❑ YoloV5 Has large I/O datasets
- ❑ gRPC uses Protobuf, **smaller data size** than REST (JSON).
- ❑ Network latency often overshadows hardware performance in user experience

Q3. Which is Better in SCS for Performance: More Instances or Increased CPU Cores?

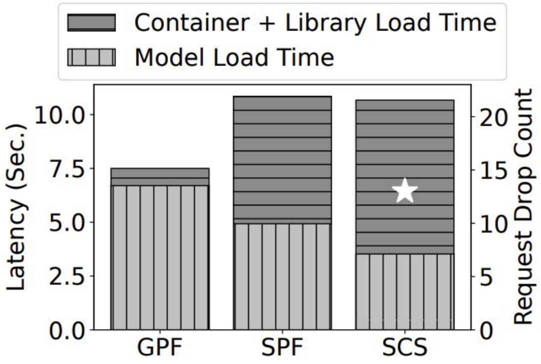
SCS: Performance with Instance & CPU Changes



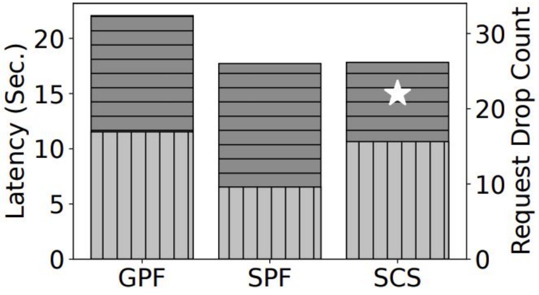
- 48 requests, processing time measured with 8GB RAM.
- Cost based on instance and CPU core count.
  - More requests per **CPU core**: similar time, longer end-to-end due to SCS queue.
  - Same instance x CPU cores: favor **more instances** over more cores per instance.

Q4. What is the Impact of Cold Start and the Need for Further Research?

Significance of Cold Start for DNN Inference



(a) MobileNetV1



(b) BERT

- GPF's shorter container load time than SPF, SCS: **lacks HTTP/gRPC server libraries** in image.
- Bigger **BERT** model size increases **container load time**
- SCS: High Cold-Start Deletion, 50% Requests Uncompleted
- Long and network latencies extend cold start time, impacting user experience.

Summary

- **Q1. How do SPF, GPF and SCS compare in performance across most DNN models?**
  - SCS most cost-effective
  - SPF faster for small models, SCS for large model
  - GPF efficient depending on overhead.
- **Q2. How does API endpoint protocol impact inference time in end-to-end response?**
  - Network latency can dominate inference time → gRPC more preferable
- **Q3. Which is Better in SCS for Performance: More Instances or Increased CPU Cores**
  - Reduced CPU cores can lengthen end-to-end response times, while using more instances proves more cost-effective for equal instance x core counts.
- **Q4. What is the Impact of Cold Start and the Need for Further Research?**
  - Loading DNN models, container images, libraries for inference services takes time and recognizing cold start issues is crucial.

Q&A

Appendix : Rest & gRPC

REST	gRPC
❑ Based on HTTP protocol: Utilizes standard web protocols and methods. ❑ Resource-oriented: Interactions revolve around resource URLs. ❑ Text-based formats: Typically uses JSON or XML for data exchange.	❑ Uses HTTP/2 protocol: Enhances performance and speed. ❑ Employs Protocol Buffers: Efficient binary serialization format. ❑ Supports streaming: Client, server, and bi-directional streaming capabilities.

Appendix : AVX512

- AVX512, short for Advanced Vector Extensions 512, is a set of instructions for Intel processors.
- It supports 512-bit wide vector operations, enhancing performance in high-performance computing and data analysis.
- Extends previous AVX and AVX2 sets, allowing more data processing in a single instruction, mainly beneficial in vectorizable operations.

Appendix : Json vs Prototuf

JSON	Prototuf
❑ Text-based format, highly readable. ❑ Widely used in web, easy to use across multiple programming languages. ❑ Larger data size and slower parsing compared to Protobuf, but highly flexible.	❑ Binary data format developed by Google, efficient in serialization. ❑ Smaller data size, faster in serialization and deserialization than JSON. ❑ Strict schema-based, less universal than JSON but excellent for performance-critical systems.

Appendix : GPF, SPF, SCS

GPF (General Purpose FaaS)	SPF (Special Purpose Faas)	SCS (serverless Container Service)
Register custom code, set memory and runtime limits	A new type of FaaS designed for specific tasks	Enables developers to deploy custom apps in containers, instance-free.
Azure Functions	Amazon SageMaker	