WoSC10 ’24, Hong Kong

INTELLIGENT OPTIMIZATION OF DISTRIBUTED PIPELINE EXECUTION IN SERVERLESS PLATFORMS: A PREDICTIVE MODEL APPROACH

Usama Benabdelkrim-Zakan, Germán T.Eizaguirre, Pedro García-López

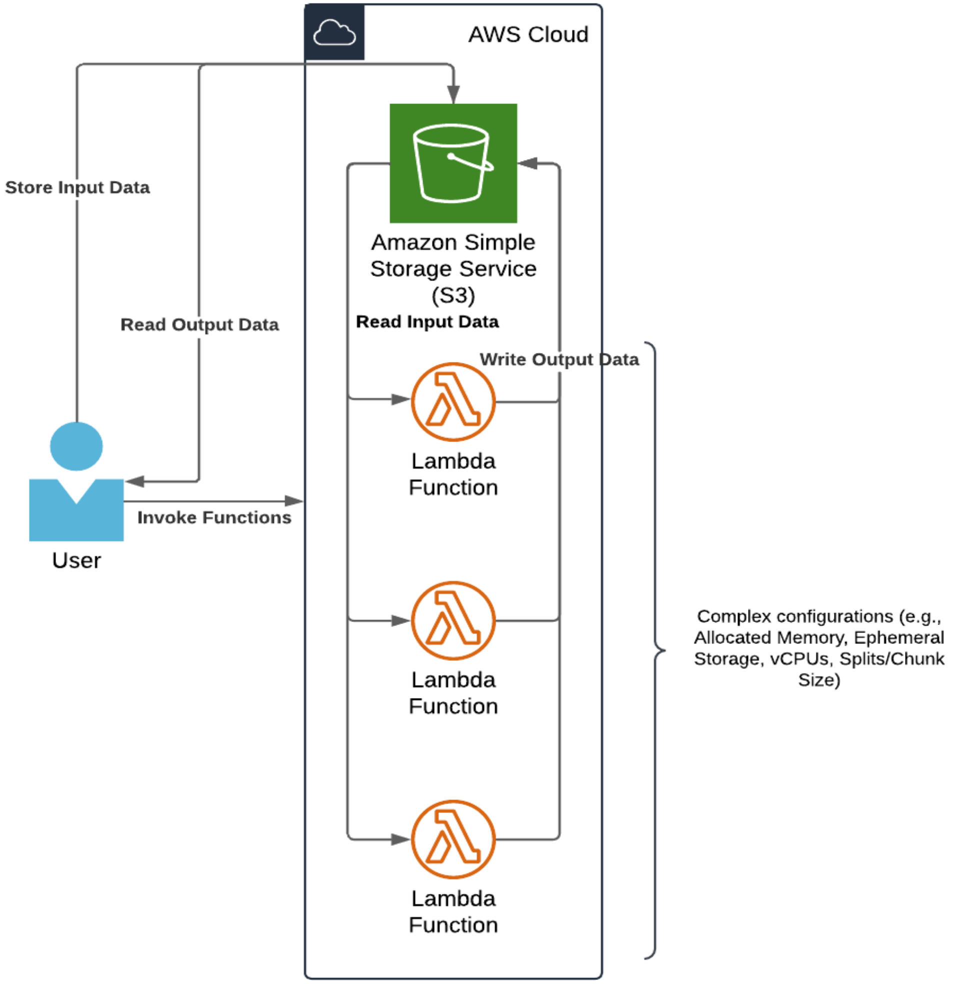

Figure 1: Serverless architecture for pipeline execution, showing AWS S3, Lambda functions, and key configuration variables (e.g., memory, vCPUs, splits).

CHALLENGES IN OPTIMIZING DISTRIBUTED PIPELINES

Serverless platforms (e.g., AWS Lambda) are popular for their scalability and low costs.

CHALLENGES

- Complex configurations impacting cost and execution time.
- Dependence on exhaustive Design Space Analysis (DSA), which is costly and slow.

Need for a predictive approach to optimize configurations efficiently.

WHAT ARE WE SOLVING?

Develop a predictive model to optimize:

- Execution time
- Operational costs

Reduce reliance on exhaustive DSA

Validate using a geospatial pipeline executed on Lithops.

STATE OF THE ART

DATAPLUG — Focuses on partition size but ignores other parameters.

SIZELESS — Predicts memory but offers limited cost and time improvements.

OUR MODEL — Simultaneous optimization of multiple parameters.

Achieves a 30% cost reduction compared to DSA.

WATER CONSUMPTION PIPELINE: USE CASE VALIDATION

DATA PREPARATION

Uploading and converting Digital Terrain Models (DTMs).

RASTER DATA INTERPOLATION

Parallel interpolation of climate variables at scale.

POTENTIAL EVAPORATION COMPUTATION

Estimating evapotranspiration from climate data.

RESULT VISUALIZATION

Generating visual representations of the analysis results.

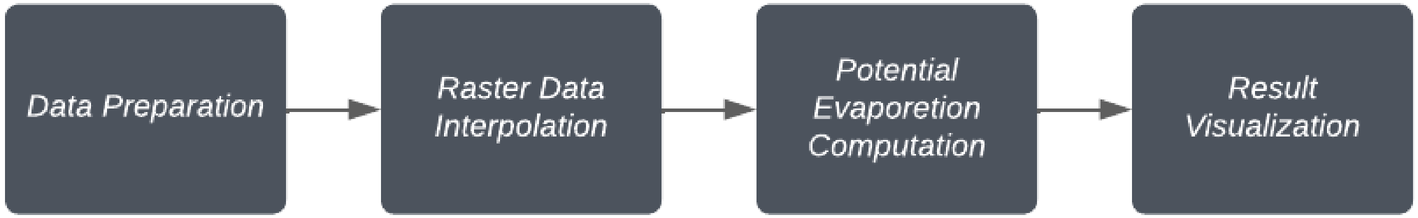

Figure 2: Water consumption pipeline stages.

OPTIMIZATION PROCESS

DATASET

148 configurations from DSA experiments.

Key parameters: splits, memory, ephemeral storage, vCPUs, input size.

PREPROCESSING

Feature engineering.

Logarithmic transformation of execution time.

Data augmentation.

MODEL

Algorithm: XGBoost.

Hyperparameter tuning: Optuna.

Figure 3: Process flow for data collection, preprocessing, training, prediction, and validation in the optimization pipeline.

FEATURE ENGINEERING

ORIGINAL PARAMETERS

Parameter	Description
num_files	Number of input files processed
splits	Number of splits (chunks) used for parallel processing
input_size_gb	Total size of the input data in gigabytes
runtime_memory_mb	Amount of memory allocated for the runtime (MB)
ephemeral_storage_mb	Temporary storage allocated for intermediate data (MB)
worker_processes	Number of worker processes running in parallel
invoke_pool_threads	Number of threads per invocation
vcpus	Number of virtual CPUs allocated

Table 1: Input Parameters Collected During DSA.

DERIVED PARAMETERS

Derived Parameter	Description
memory_per_file	Memory allocated per file processed (MB)
storage_per_file	Temporary storage per file (MB)
vcpus_per_file	vCPUs allocated per file
files_per_vcpu	Number of files processed per vCPU
size_per_file	Size of each file (GB)
memory_per_gb	Memory allocated per GB of input size
vcpus_per_gb	vCPUs allocated per GB of input size
storage_per_gb	Temporary storage per GB of input size (MB)
threads_per_worker	Threads running per worker process
memory_per_thread	Memory allocated per thread (MB)
vcpus_per_thread	vCPUs allocated per thread
memory_per_thread_vcpus_ratio	Ratio of memory to vCPUs per thread

Table 2: Derived Parameters from Feature Engineering.

KEY RESULTS

MAE REDUCTION

75.34% vs. Baseline (Average).

69% vs. Linear Regression.

COST REDUCED BY 30% COMPARED TO DSA

INVESTMENT RECOVERY IN JUST 2 MONTHS

Assuming a rate of 10 executions per day.

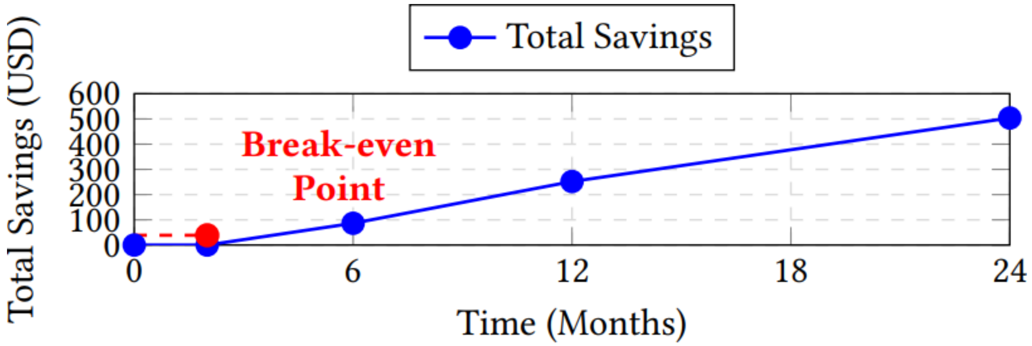

Figure 4: Break-even point graph showing the recovery of investment within 2 months.

MODEL VALIDATION

Unseen Configurations

- Predicted optimal duration: ~ 195s (real: 184s).

Residual Analysis

- Symmetrical residuals indicate low bias.

Learning Curve

- Demonstrates strong generalization with limited data.

Model	MAE (s)	Avg. MAE (CV) (s)	MAPE (%)	R^2
XGBoost	29.81	34.20	8.72%	0.8802
Baseline (Average)	120.90	-	-	-
Linear Regression	97.02	96.62	28.73%	0.3380
PCA + Linear Regression	97.70	92.03	29.04%	0.3240

Table 3: Comparison of Models.

COMPARISON: REAL VS. PREDICTED

XGBoost predictions align closely with actual values.

Superior to simpler methods like Linear Regression or PCA-based models.

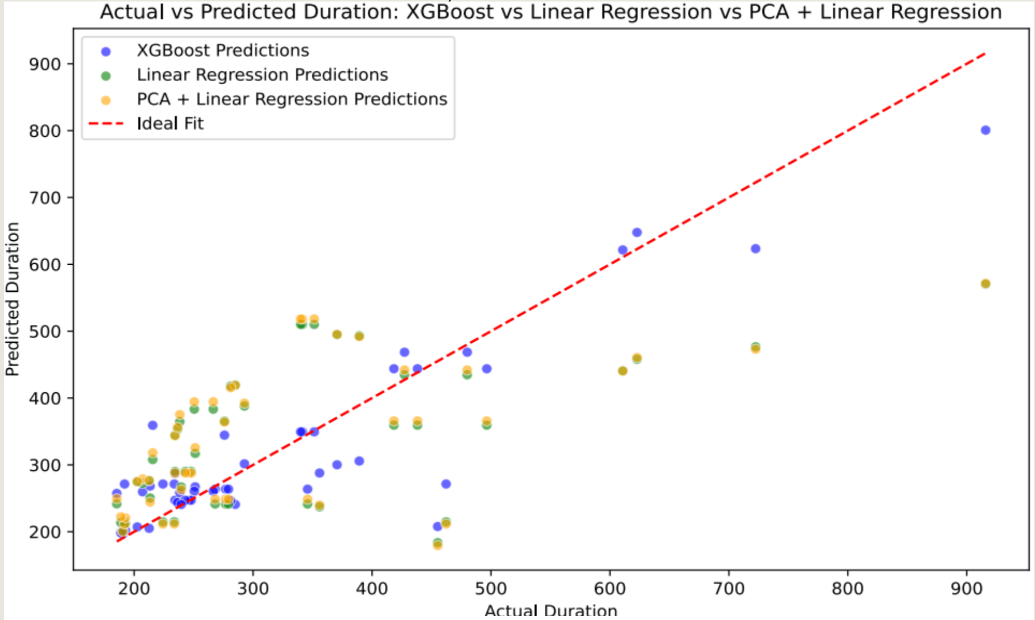

Figure 5: Comparison of actual vs. predicted duration for various models, highlighting the performance of XGBoost against simpler regression methods.

KEY TAKEAWAYS

PREDICTIVE MODEL EFFECTIVELY OPTIMIZES SERVERLESS PIPELINES

ACHIEVED:

- Up to 79.9% reduction in execution time.
- ~30% cost savings.

APPLICABLE ACROSS SERVERLESS PLATFORMS (AWS LAMBDA, AZURE, GOOGLE CLOUD)

NEXT STEPS

IMPROVE ACCURACY WITH LARGER DATASETS.

EXPLORE ADVANCED ARCHITECTURES (E.G., NEURAL NETWORKS).

VALIDATE ON DIVERSE PIPELINES AND SERVERLESS PLATFORMS.

ACKNOWLEDGMENTS AND Q&A

FUNDED BY:

- MINISTRY OF ECONOMIC AFFAIRS AND DIGITAL TRANSFORMATION (CLOUDLESS PROJECT).
- HORIZON EUROPE (CLOUDSKIN PROJECT).

SPECIAL THANKS TO REVIEWERS AND COLLABORATORS.

CORRESPONDING AUTHOR:

- USAMA.BENABDELKRIM@URV.CAT

THANK YOU

Usama Benabdelkrim-Zakan

- usama.benabdelkrim@urv.cat

Germán T.Eizaguirre

- germantelmo.eizaguirre@urv.cat

Pedro García-López

- pedro.garcia@urv.cat