

SparkAIBench: A Benchmark to Generate AI Workloads on Spark

Presenter: Liu Zifeng

Beijing Institute of Technology



Outline

- Background and Motivation
- *SparkAIBench*
 - *Overview*
 - *Process of Workload Generation*
 - *Available AI Algorithms*
 - *Expression of Workload Generation Requirement*
- Use Case
- Conclusion



Outline

- Background and Motivation
- SparkAIBench
 - *Overview*
 - *Process of Workload Generation*
 - *Available AI Algorithms*
 - *Expression of Workload Generation Requirement*
- Use Case
- Conclusion



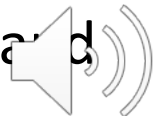
AI workloads in the Cloud

Recent years, distributed machine (deep) learning workloads, referred to as AI workloads, are rapidly becoming prevalent and potential applications in cloud computing.



Existing problems

- There is a lack of workload in the field of artificial intelligence.
 - The major efforts on generating workloads today do not focus on AI domain. And there is no study which is able to automatically generate user customized AI workloads.
 - Workloads generation is one of the most important aspect in benchmarking, generating in a manual manner is quite complicated.
- Example
 - DRL-based scheduler mostly trains agent through the cluster traces generated by running workloads whose characteristics are configured manually due to the lack of frameworks that enable generating diverse and customized user workloads automatically.



Outline

- Background and Motivation
- SparkAI Bench
 - *Overview*
 - *Process of Workload Generation*
 - *Available AI Algorithms*
 - *Workload Generation Requirement*
- Use Case
- Conclusion



SparkAIBench

■ Overview

- This paper we present a benchmark to generate AI workloads, which supports a variety of AI algorithms, changeable input data size, as well as parametric method for submission.



SparkAIBench

■ Overview

■ The contributions

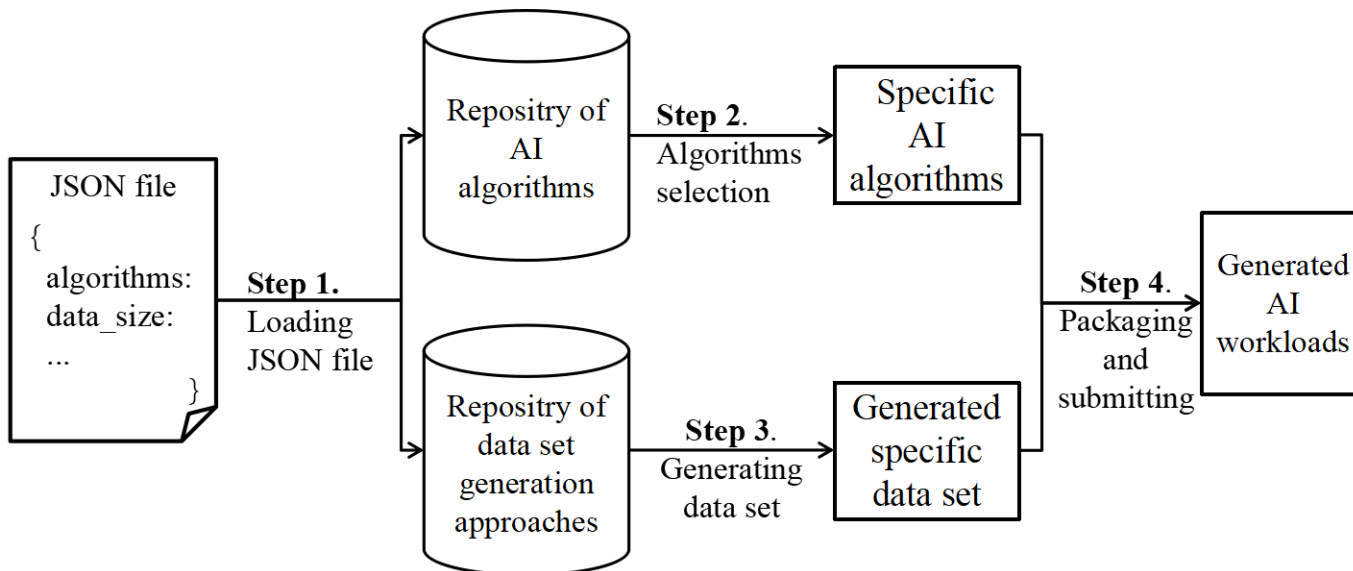
- A user customized and automatic AI workloads generator
- A use case to illustrate how SparkAIBench works in a real job scheduling optimization scenario.



SparkAIBench

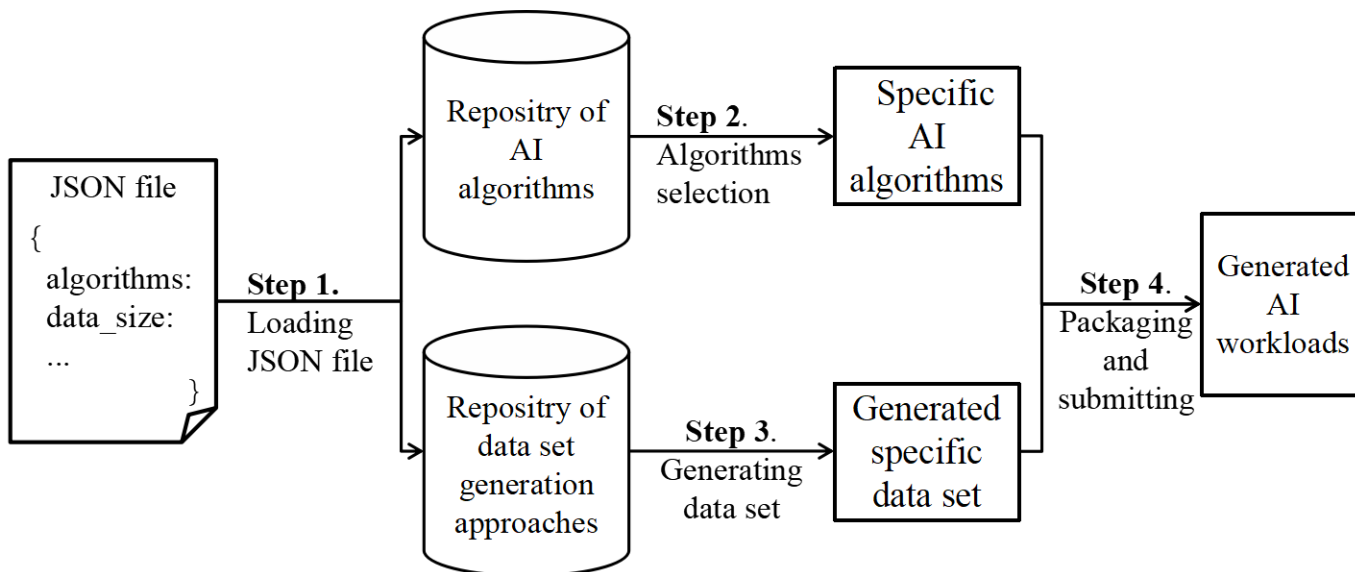
■ Process of Workload Generation

- 1. reading a requirement of AI workloads generation from a JSON file, SparkAIBench is able to know how many workloads should be generated.



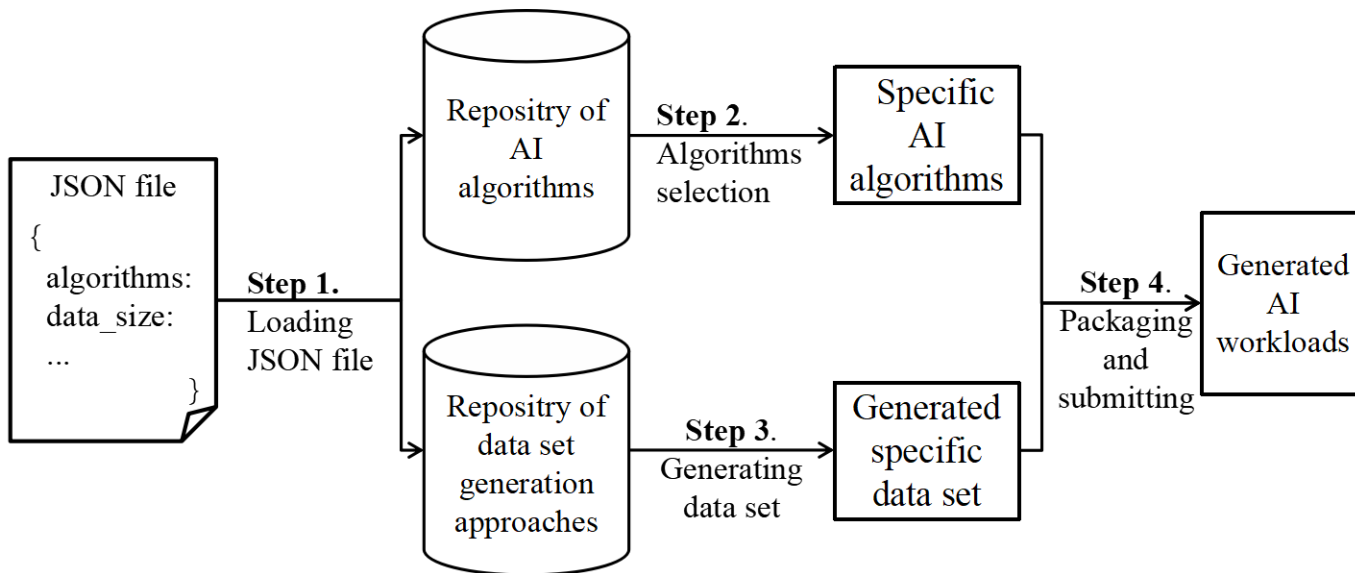
SparkAIBench

- 2. select specific machine learning algorithms within Spark MLlib or BigDL according to value of “*algorithms*”
- 3. according to selected algorithms and the value of “*data_size*”, SparkAIBench chooses corresponding data generation methods to obtain the training data sets and send them into HDFS.



SparkAIBench

- 4. package the above algorithms into an assembly jar and put it into YARN-based Spark platform as an application



SparkAIBench

■ Available AI Algorithm

Framework	Algorithm	Domain	Data generator
Spark <u>MLlib</u>	Linear Regression	Regression Analysis	SELF
	LDA	Text Mining	SELF
	Bayes	Classification	<u>LibSVM</u>
	SVM	Classification	<u>LibSVM</u>
	<u>FP-Growth</u>	<u>Frequent Itemset Mining</u>	BDGS
	<i>k</i> -means	Clustering	BDGS
	ALS	Recommendation	BDGS
<u>BigDL</u>	<u>LeNet</u>	Image Classification	MINST
	Inception	Image Classification	<u>ImageNet</u>
	VGG	Image Classification	CIFAR-10
	<u>ResNet</u>	Image Classification	CIFAR-10
	RNN	Natural Language Processing	Tiny Shakespeare Texts corpus
	Auto-encoder	Image Classification	MINST



SparkAIBench

■ Workload Generation Requirement

- In order to flexibly and controllably represent a user requirement of AI workloads generation, we transform it into a JSON object with several configurable parameters shown in Table (i.e. keys of such JSON object), and insert the object into a JSON file.

Name	Description
<i>interval</i>	Submission interval among different algorithms.
<i>algorithms</i>	An integer array with a size of 13 represents which algorithms are selected.
<i>data_size</i>	An integer array with a size of 13 represents the data scale of each selected algorithm.
<i>queue</i>	The job queue where the algorithms are submitted to.
<i>priority</i>	The job priority of each algorithm.



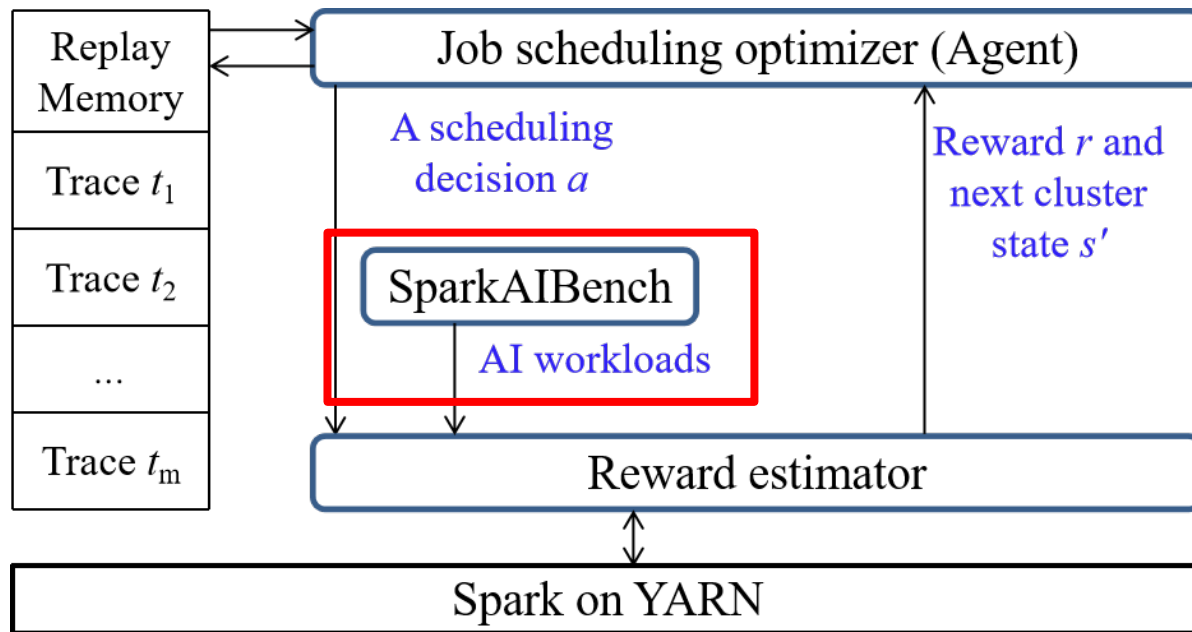
Outline

- Background and Motivation
- SparkAI Bench
 - *Overview*
 - *Process of Workload Generation*
 - *Available AI Algorithms*
 - *Expression of Workload Generation Requirement*
- Use Case
- Conclusion



Use Case

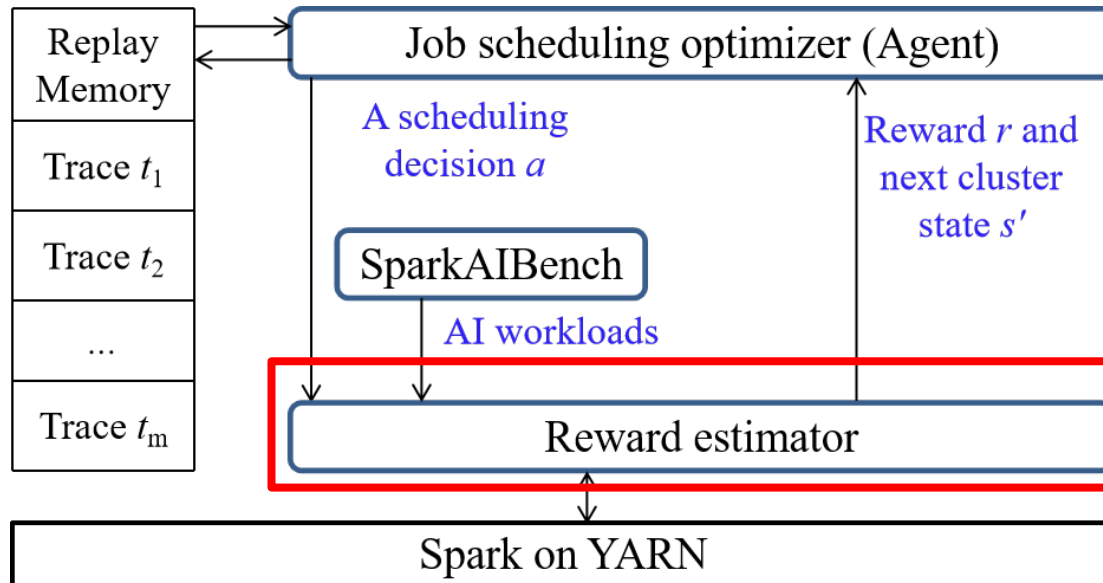
- a DRL-based job scheduling optimizer
 - the aim of SparkAIBench in this scenario is to generate various AI workloads for training the *job scheduling optimizer* (agent).



Use Case

■ Reward Estimator

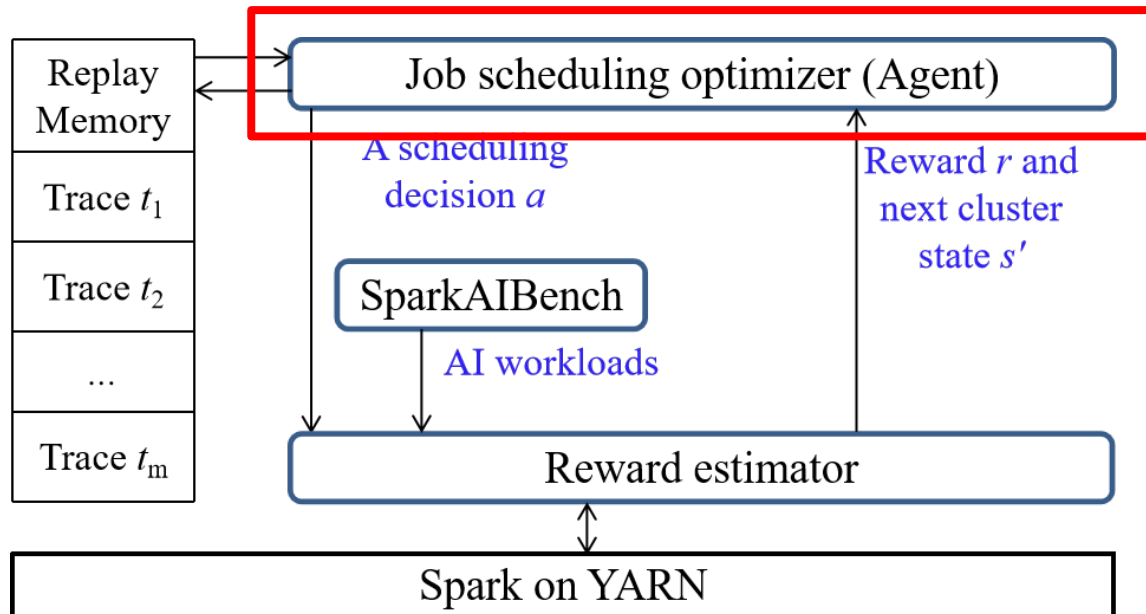
- The estimator is regarded as a reward function used in DRL mechanism. If carrying out a scheduling decision makes a lower average job latency, it means the scheduling decision improves cluster's performance, and vice versa.



Use Case

■ Job Scheduling Optimizer (Agent)

- In DRL-based optimizer (agent), two neural networks are introduced, which both take expected accumulated reward as output and with the same model structure.



Use Case

■ Proposing Requirements of AI Workloads Generation

SERVICES

Generate AI workloads you want


Input the queue's name:

Priority:

Interval(s):

Algorithms selection
and
Data size setting

<input checked="" type="checkbox"/>	1000
<input type="checkbox"/>	LDA
<input type="checkbox"/>	Bayes
<input checked="" type="checkbox"/>	2
<input type="checkbox"/>	FP-Growth
<input checked="" type="checkbox"/>	1500
<input type="checkbox"/>	ALS
<input type="checkbox"/>	LeNet
<input type="checkbox"/>	Inception
<input type="checkbox"/>	VGG
<input type="checkbox"/>	ResNet
<input type="checkbox"/>	RNN
<input type="checkbox"/>	Auto-encoder



Outline

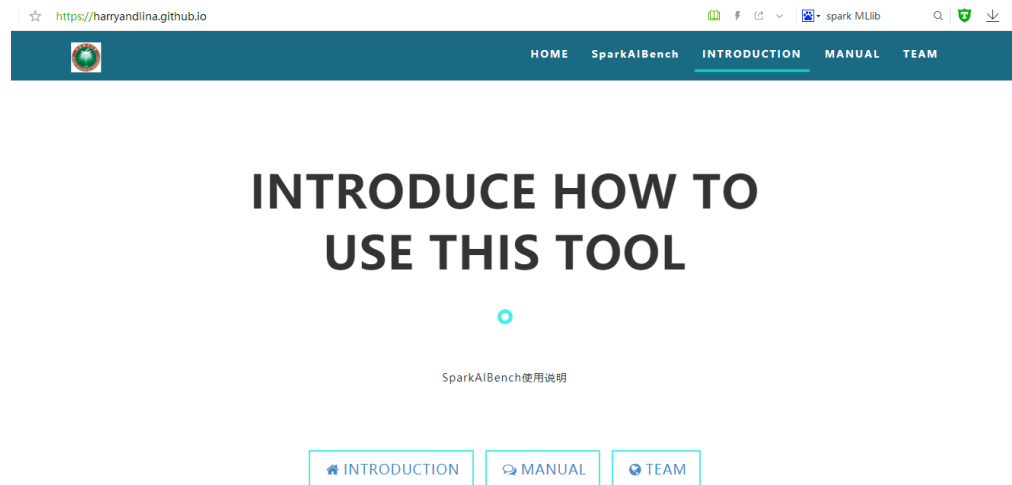
- Background and Motivation
- SparkAIBench
 - *Overview*
 - *Process of Workload Generation*
 - *Available AI Algorithms*
 - *Expression of Workload Generation Requirement*
- Use Case
- Conclusion



Conclusion

■ SparkAIBench

- a user customized benchmark, SparkAIBench, with the ability of generating various AI workloads through a configurable user requirement file.
- Project Homepage
 - User manual: <https://harryandlina.github.io/>



Thanks

Presenter: Liu Zifeng

1217750686@qq.com

Beijing Institute of Technology

