

Using Apache Spark for scientific research

Basic Concepts and Scientific Examples

Sungryong Hong
KASI, 6/2/2022

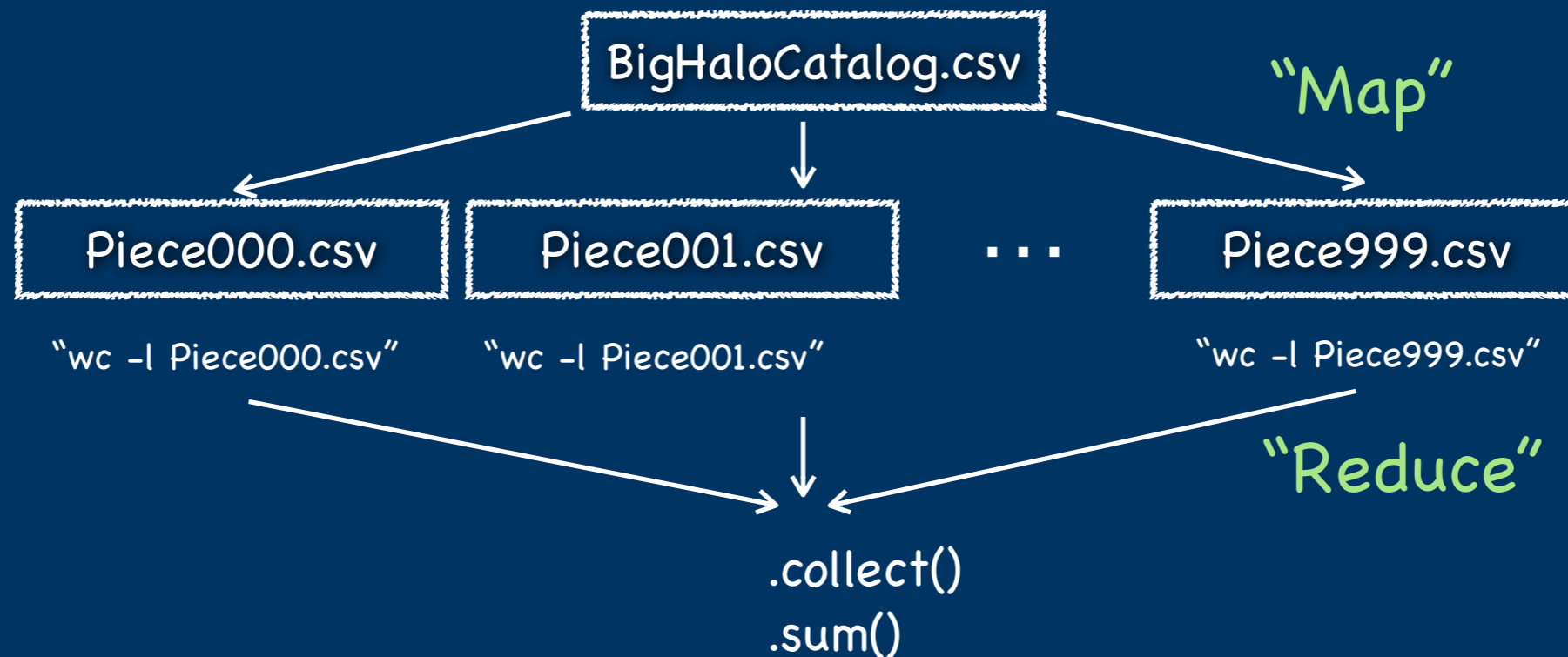
What is “Big” Data in Our Real Life?

Even a very simple calculation is challenging in Big Data.

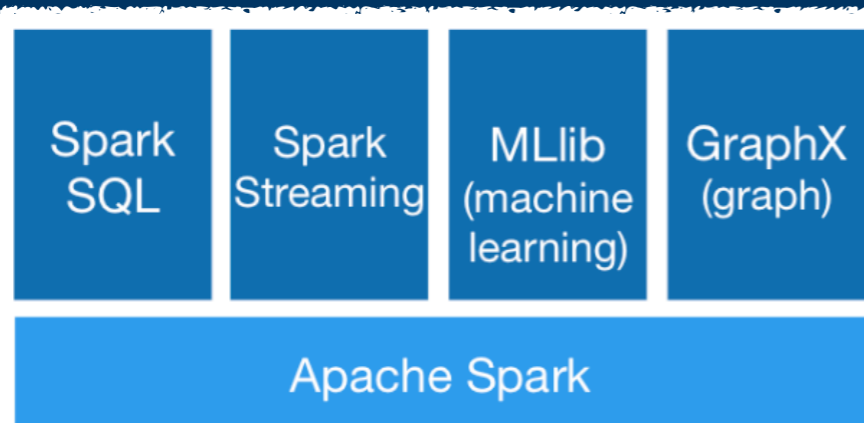
Q: Find the total number of halos in “BigHaloCatalog.csv” (1TB)

A: bash> wc -l BigHaloCatalog.csv [Enter]

Segmentation Fault...



Two common Big Data Platforms



DASK

Dask natively scales Python

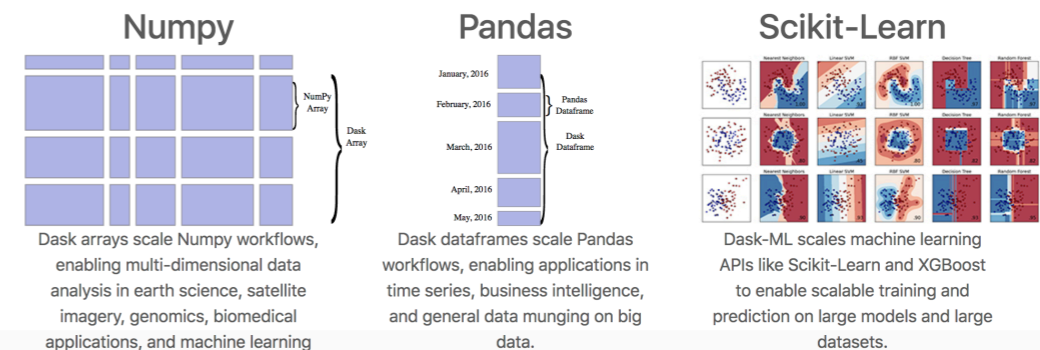
Dask provides advanced parallelism for analytics, enabling performance at scale for the tools you love

[Learn More](#) [Try Now »](#)

Integrates with existing projects

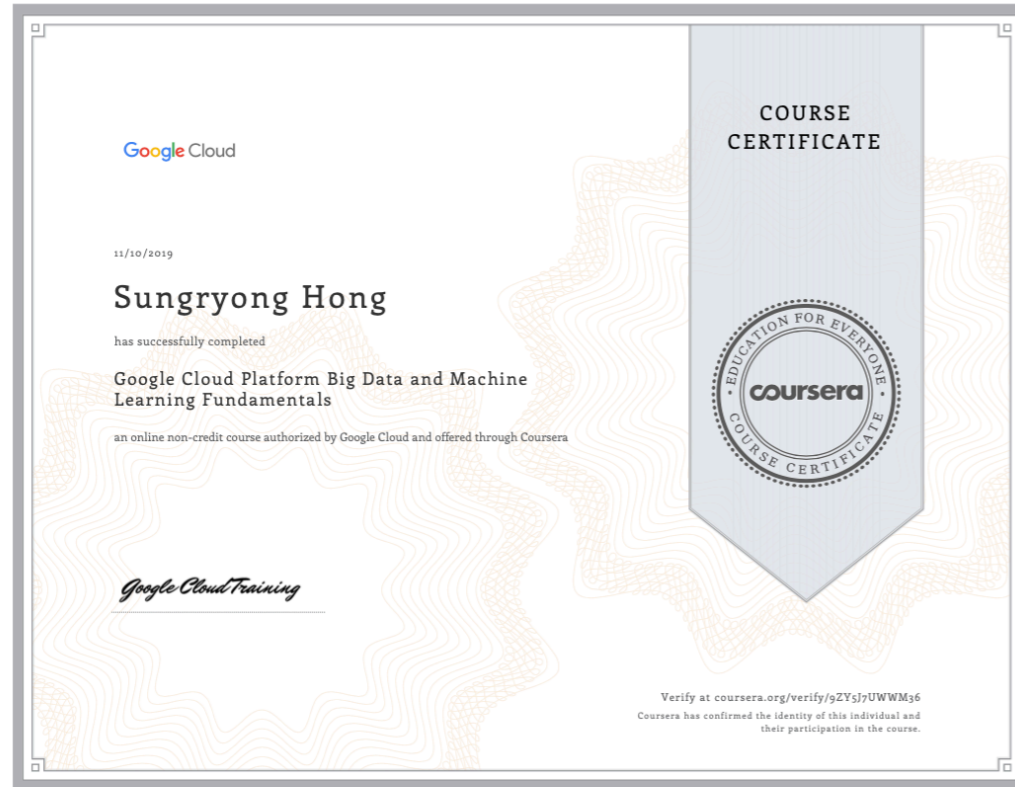
BUILT WITH THE BROADER COMMUNITY

Dask is open source and freely available. It is developed in coordination with other community projects like Numpy, Pandas, and Scikit-Learn.



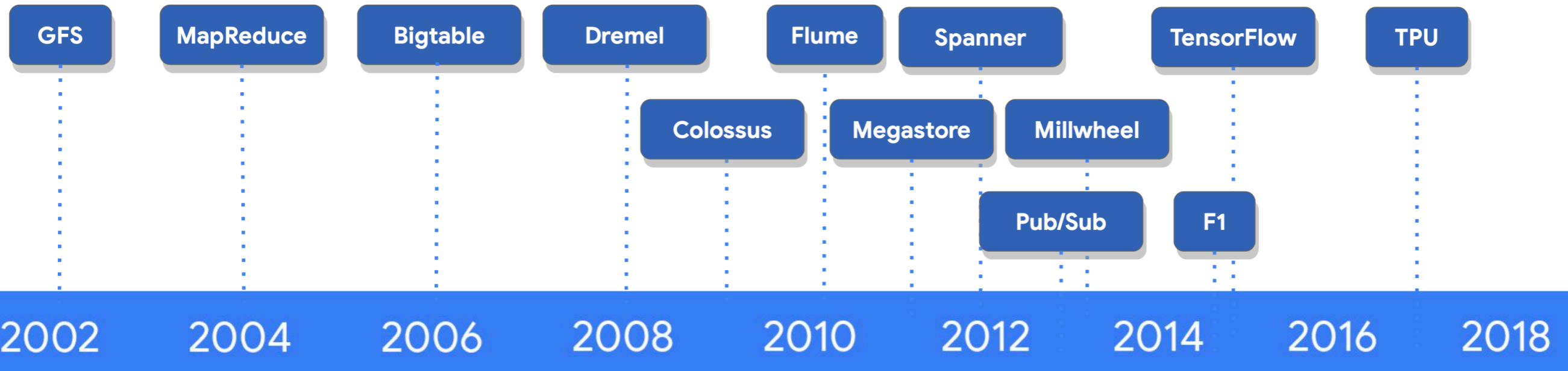
Basic Prerequisites are
Python Data Science Stacks:
numpy, scipy, pandas, scikit-learn

Quick History of Big Data Techs (feat. GCP)

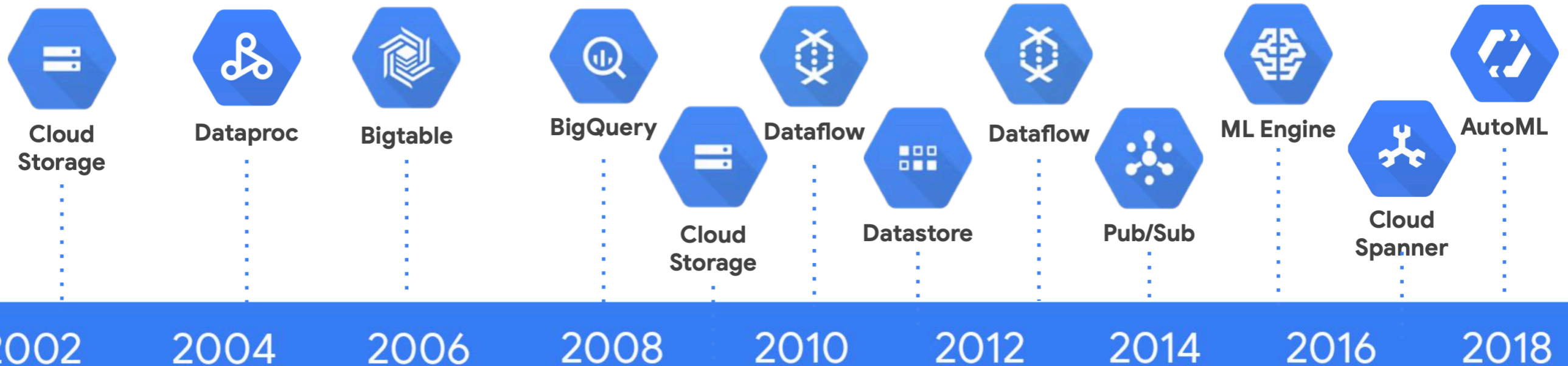


Quick History of Big Data Techs (feat. GCP)

Google invented new data processing methods as it grew



Google Cloud opens up that innovation and infrastructure to you



What is Apache Spark?

“Apache Spark is a **multi-language** engine for executing data engineering, data science, and machine learning on single-node machines or **clusters**.”

Simple.
Fast.
Scalable.
Unified.

Key features



Batch/streaming data

Unify the processing of your data in batches and real-time streaming, using your preferred language: Python, SQL, Scala, Java or R.



SQL analytics

Execute fast, distributed ANSI SQL queries for dashboarding and ad-hoc reporting. Runs faster than most data warehouses.



Data science at scale

Perform Exploratory Data Analysis (EDA) on petabyte-scale data without having to resort to downsampling



Machine learning

Train machine learning algorithms on a laptop and use the same code to scale to fault-tolerant clusters of thousands of machines.

What is Apache Spark?



Basic Prerequisites are
Python Data Science Stacks:
numpy, scipy, pandas, scikit-learn

pandas

vs.

Spark DataFrame, koalas, pyspark.pandas

```
import pyspark.pandas as ps
```

scikit-learn

vs.

MLlib, SparkML, MMLSpark(SynapseML)

Immutable Data and Lazy Evaluation

pandas vs. Spark DataFrame

mutable variable의 간단한 예를 들면,

```
a = 1
b = 1
a = a + 1
a = a + b
b = b + a
c = a + b
c = c + a
print(c)
```

immutable inputs “a”와 “b”를 이용해서 최종 c를 출력하는 방식을 찾아야하는데 ..
위의 방정식을 immutability를 지키면서 따라가면

우선, 첫 4줄까지 실행하면

$a_cache = b + (a + 1)$

다섯 번째 줄은

$b_cache = b + a_cache$

여섯 번째 줄은

$c_cache = a_cache + b_cache$

일곱 번째 줄은

$c = c_cache + a_cache$

이렇게 됩니다.

그래서 최종 c 값은

$c = (b + (a + 1) + (b + (b + (a + 1))) + (b + (a + 1)))$

이렇게 됩니다 ..

Immutable Input > Directed Acyclic Graph (DAG) > Lazy Evaluated Result

Creating Apache Spark Clusters

Spark and Cluster Managers

Spark Standalone Cluster



Apache Mesos



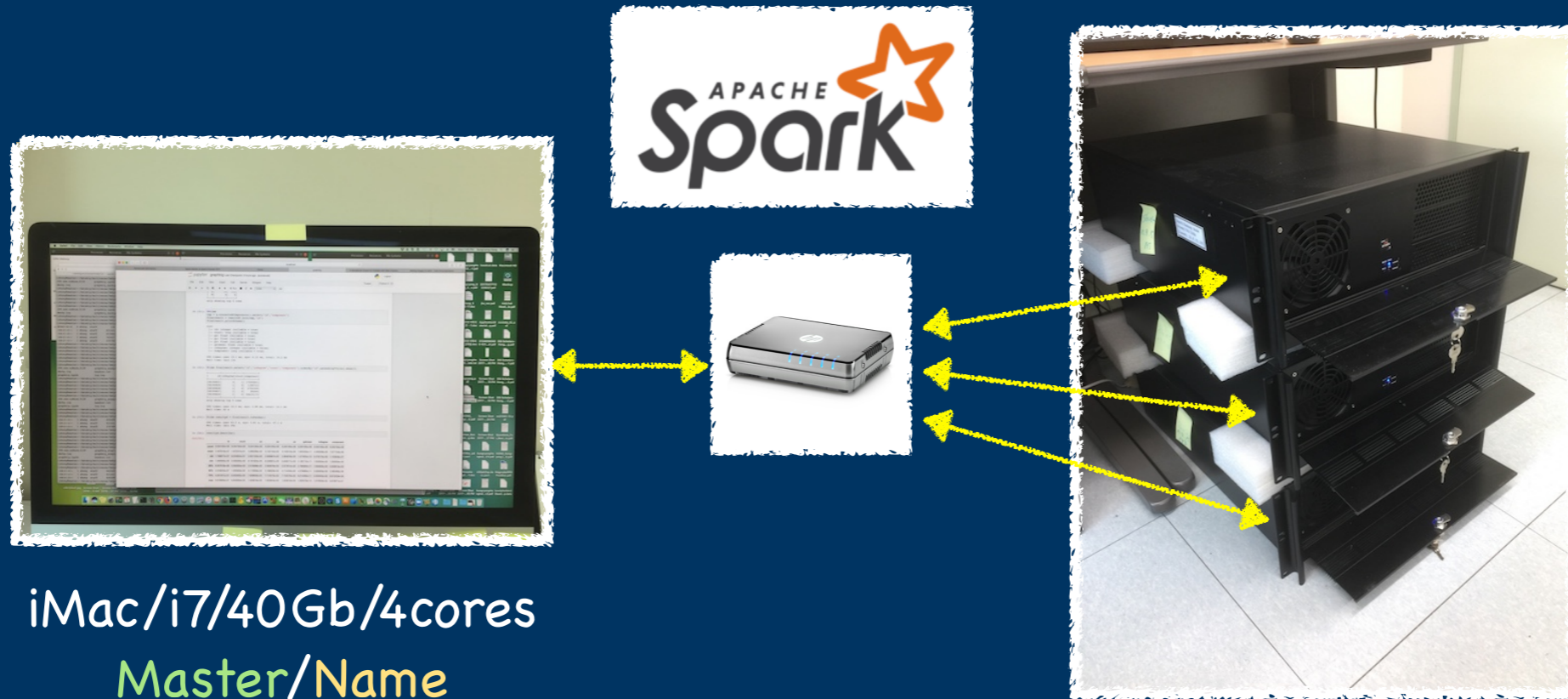
Hadoop YARN



Kubernetes



Creating Apache Spark Clusters



iMac/i7/40Gb/4cores
Master/Name

Multiverse Simulation, Horizon
Run 4, SDSS, and CMB data

3 x Ubuntu/Ryzen/64GB/8cores
Slave/Data



Creating Apache Spark Clusters

Project



Project / Cluster Infra / Cluster Template

API Access

Compute



Cluster Template

Cluster Infra



KASI Clusters

KASI Cluster Templates

Resource Types

Template Versions

Template Generator

Container Infra



Network



Volumes



Object Store



Share



Rating



Identity



Management



Filter

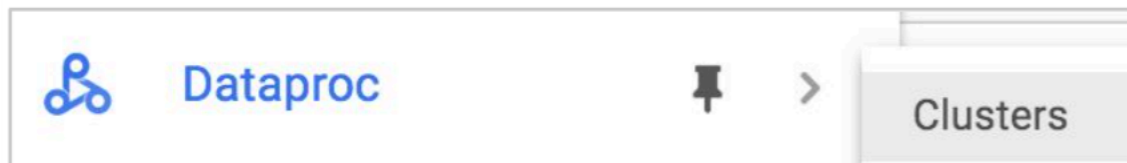


Displaying 13 items

Name	Size	Updated Time	Actions
KASI-SingleVM	4413	Wed Mar 23 11:06:15 2022	Launch Cluster Stack
KASI-OpenMPI-Cluster	11306	Fri Mar 04 15:41:36 2022	Launch Cluster Stack
KASI-OpenMPI-Cluster-Slurm	15872	Fri Mar 04 15:41:12 2022	Launch Cluster Stack
KASI-Dask-Cluster	12964	Wed Feb 16 18:37:55 2022	Launch Cluster Stack
KASI-Celery-Cluster	13153	Wed Feb 16 18:38:11 2022	Launch Cluster Stack
KASI-Airflow-Cluster-1	17661	Wed Feb 16 18:38:26 2022	Launch Cluster Stack
KASI-Airflow-Cluster-2(Dask)	18825	Wed Feb 16 18:38:44 2022	Launch Cluster Stack
KASI-Airflow-Cluster-3(Celery)	20140	Wed Feb 16 18:38:58 2022	Launch Cluster Stack
KASI-DB-Cluster-1(MongoDB)	23302	Wed Feb 16 18:39:51 2022	Launch Cluster Stack
KASI-DB-Cluster-2(ClickHouse)	11006	Wed Feb 16 18:40:04 2022	Launch Cluster Stack
KASI-Spark-Cluster(stand-alone)	16505	Wed Feb 16 18:40:22 2022	Launch Cluster Stack
KASI-Spark-Cluster(hadoop)	19906	Wed Feb 16 18:40:36 2022	Launch Cluster Stack
devstack-test-nfs	11170	Mon Feb 14 17:26:42 2022	Launch Cluster Stack

Displaying 13 items

Creating Apache Spark Clusters



← Create a cluster

Name [?]
tax-report-processing

Zone [?]
us-east1-b

Master node
Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type [?] Cluster mode [?]
n1-standard-4 (4 vCPU, 15.0 GB ... Standard (1 master, N workers)

Primary disk size (minimum 10 GB) [?]
500 GB

Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.

Machine type [?] Nodes (minimum 2) [?]
n1-standard-4 (4 vCPU, 15.0 GB ... 2

Primary disk size (minimum 10 GB) [?] Local SSDs (0-8) [?]
500 GB 0 x 375 GB

YARN cores [?] YARN memory [?]
8 24.0 GB

```
gcloud dataproc clusters create mycluster \  
  --initialization-actions gs://mybucket/init-actions/my_init.sh \  
  --initialization-action-timeout 3m
```

Scientific Examples

Let's see some Jupyter Notebooks

Example: Massive Graph Measurements

Monthly Notices

of the

ROYAL ASTRONOMICAL SOCIETY

MNRAS **493**, 5972–5986 (2020)

Advance Access publication 2020 February 27



doi:10.1093/mnras/staa566

Constraining cosmology with big data statistics of cosmological graphs

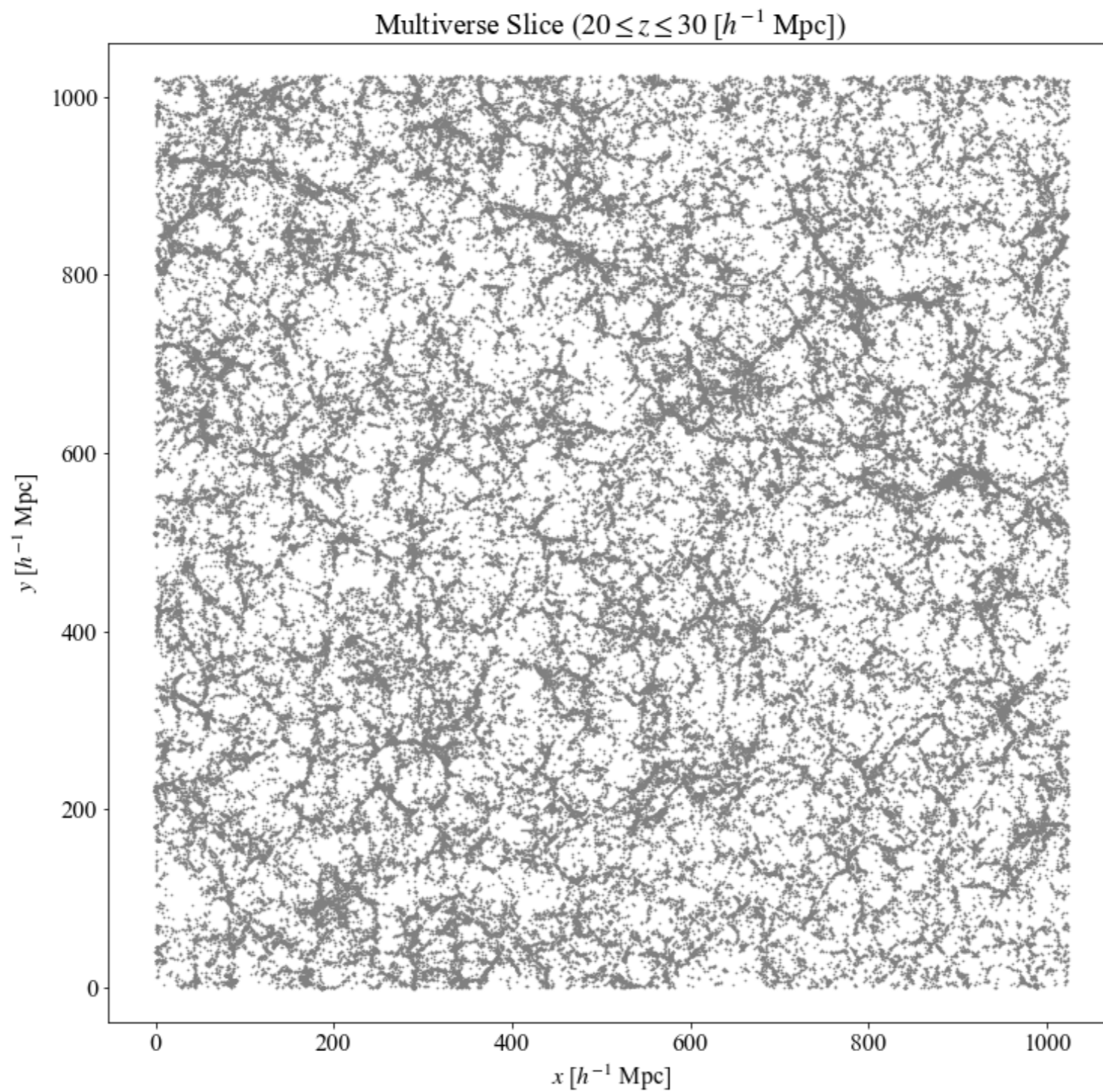
Sungryong Hong^{1,2★}, Donghui Jeong,³ Ho Seong Hwang^{2,4★}, Juhan Kim,⁵

ABSTRACT

By utilizing large-scale graph analytic tools implemented in the modern big data platform, APACHE SPARK, we investigate the topological structure of gravitational clustering in five different universes produced by cosmological N -body simulations with varying parameters: (1) a WMAP 5-yr compatible Λ CDM cosmology, (2) two different dark energy equation of state variants, and (3) two different cosmic matter density variants. For the big data calculations, we use a custom build of standalone Spark/Hadoop cluster at Korea Institute for Advanced Study and Dataproc Compute Engine in Google Cloud Platform with sample sizes ranging from 7 to 200 million. We find that among the many possible graph-topological measures, three simple ones: (1) the average of number of neighbours (the so-called average vertex degree) α , (2) closed-to-connected triple fraction (the so-called transitivity) τ_{Δ} , and (3) the cumulative number density $n_{s \geq 5}$ of subgraphs with connected component size $s \geq 5$, can effectively discriminate among the five model universes. Since these graph-topological measures are directly related with the usual n -points correlation functions of the cosmic density field, graph-topological statistics powered by big data computational infrastructure opens a new, intuitive, and computationally efficient window into the dark Universe.

Key words: methods: numerical – methods: statistical – large-scale structure of Universe – cosmology: theory.

Example: Massive Graph Measurements





Example: Massive Graph Measurements

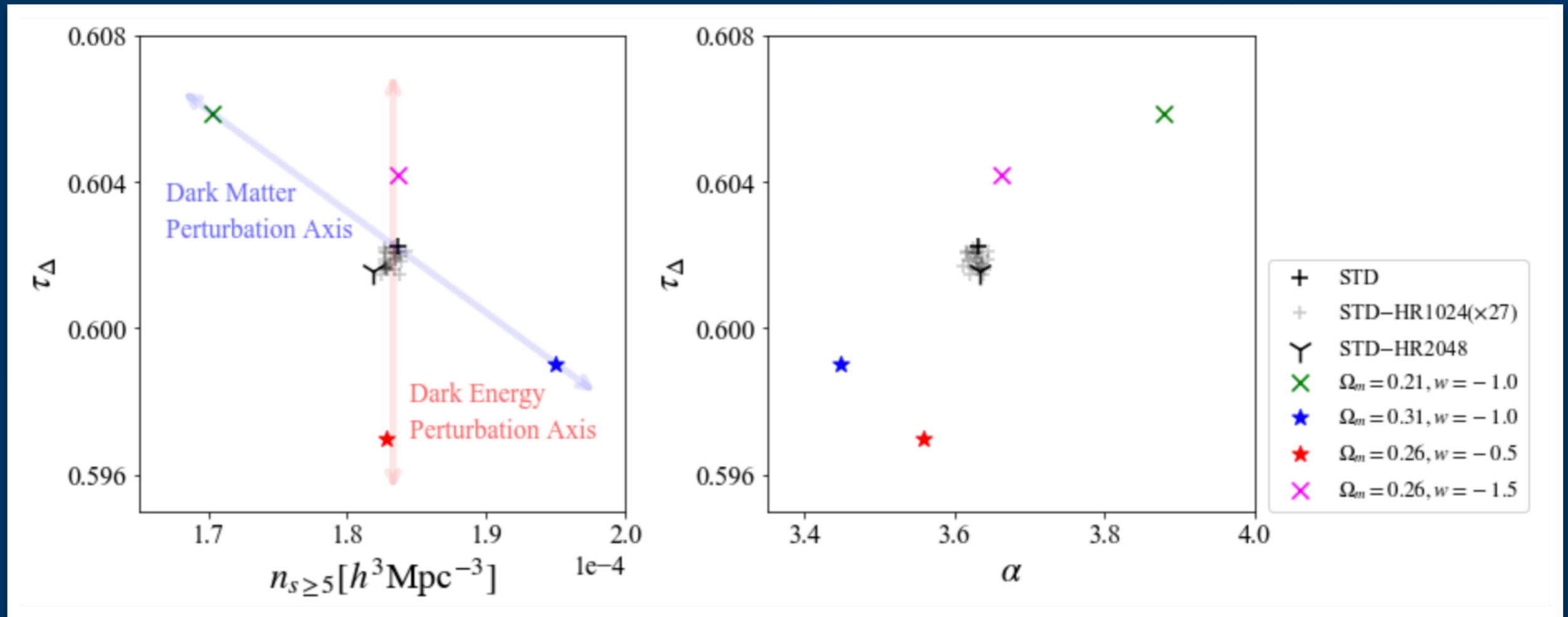
Table 2. Sample Selections

Name	Multiverses	Equal Mass Cut Sample		Equal Abundance Sample ^a	
	Cosmological Parameters	N_h	$M_{cut}(M_\odot)$	N_h	$M_{min}(M_\odot)$
STD	$\Omega_m = 0.26, w = -1.0$	7,086,717	5.00×10^{11}	7,086,717	5.05×10^{11}
DE1	$\Omega_m = 0.26, w = -0.5$	7,806,135	5.00×10^{11}	7,086,717	5.59×10^{11}
DE2	$\Omega_m = 0.26, w = -1.5$	6,886,870	5.00×10^{11}	7,086,717	4.87×10^{11}
DM1	$\Omega_m = 0.31, w = -1.0$	8,595,923	5.00×10^{11}	7,086,717	6.24×10^{11}
DM2	$\Omega_m = 0.21, w = -1.0$	5,579,491	5.00×10^{11}	7,086,717	3.86×10^{11}
STD-HR	Horizon Run [†]	206,140,716	5.00×10^{11}	206,140,716	5.05×10^{11}

Table 1. Hardware Configurations for the Spark Clusters[†]

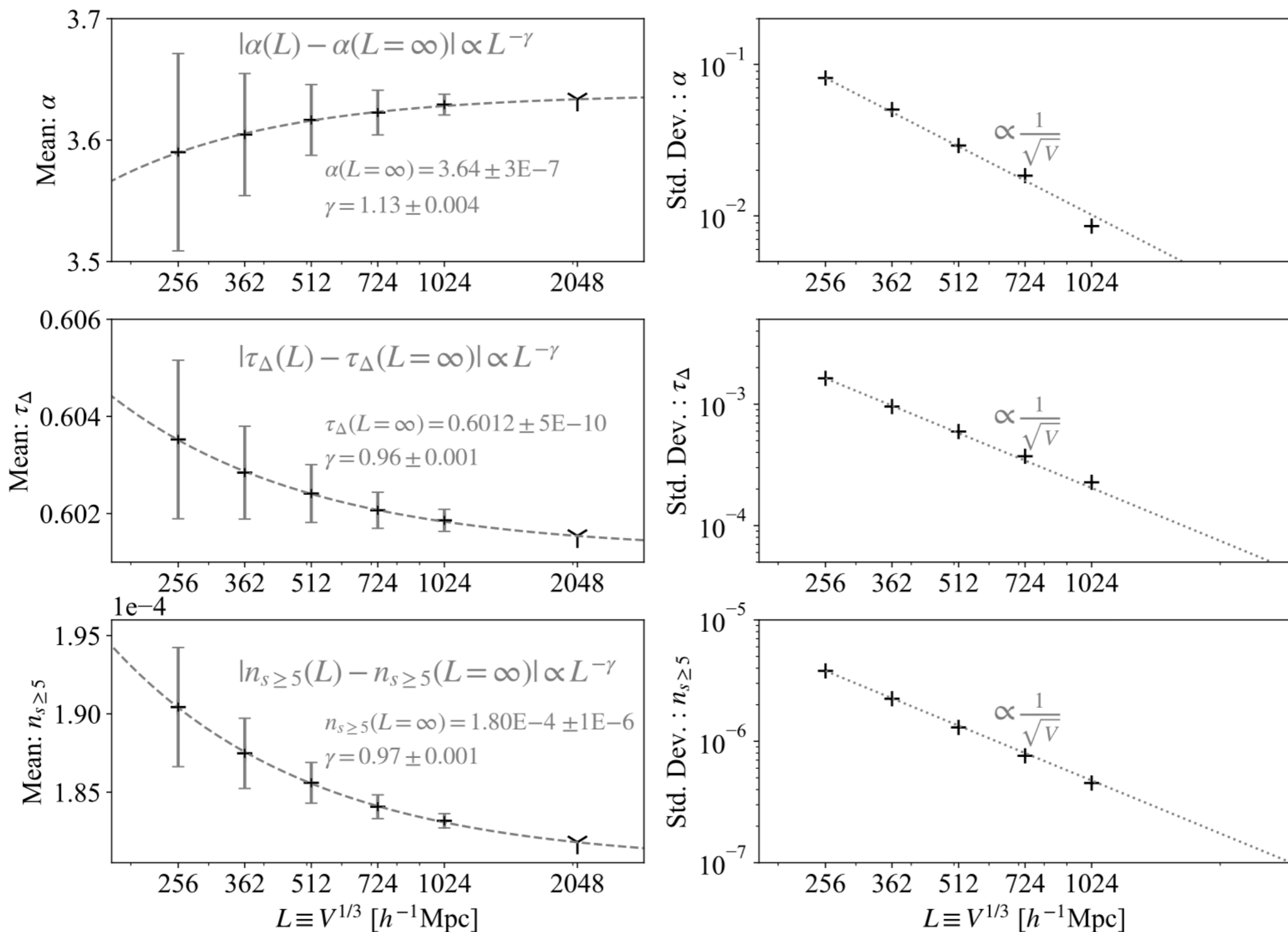
Cluster Name	Driver Node		Worker Node		
	vCPUs [†]	Memory	vCPUs [†]	Memory	n Workers [†]
KIAS Standalone ^a	4	32GB	16	52GB	3
Google Cloud Dataproc ^b	16	104GB	32	208GB	5

Example: Massive Graph Measurements

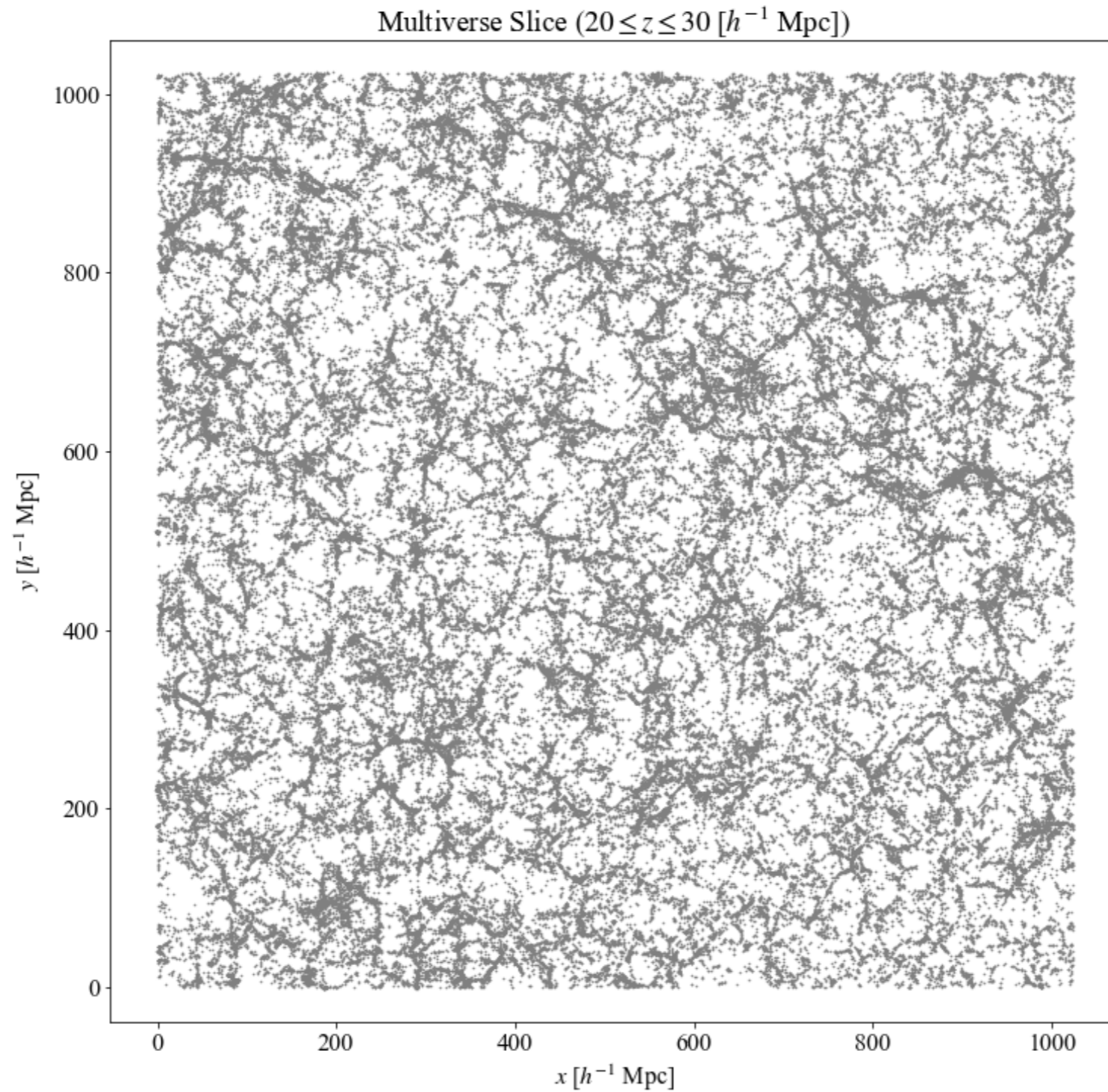


STD-HR2048: 57 millions halos with 206 millions connections
I paid \$30 for this single point.

Example: Massive Graph Measurements



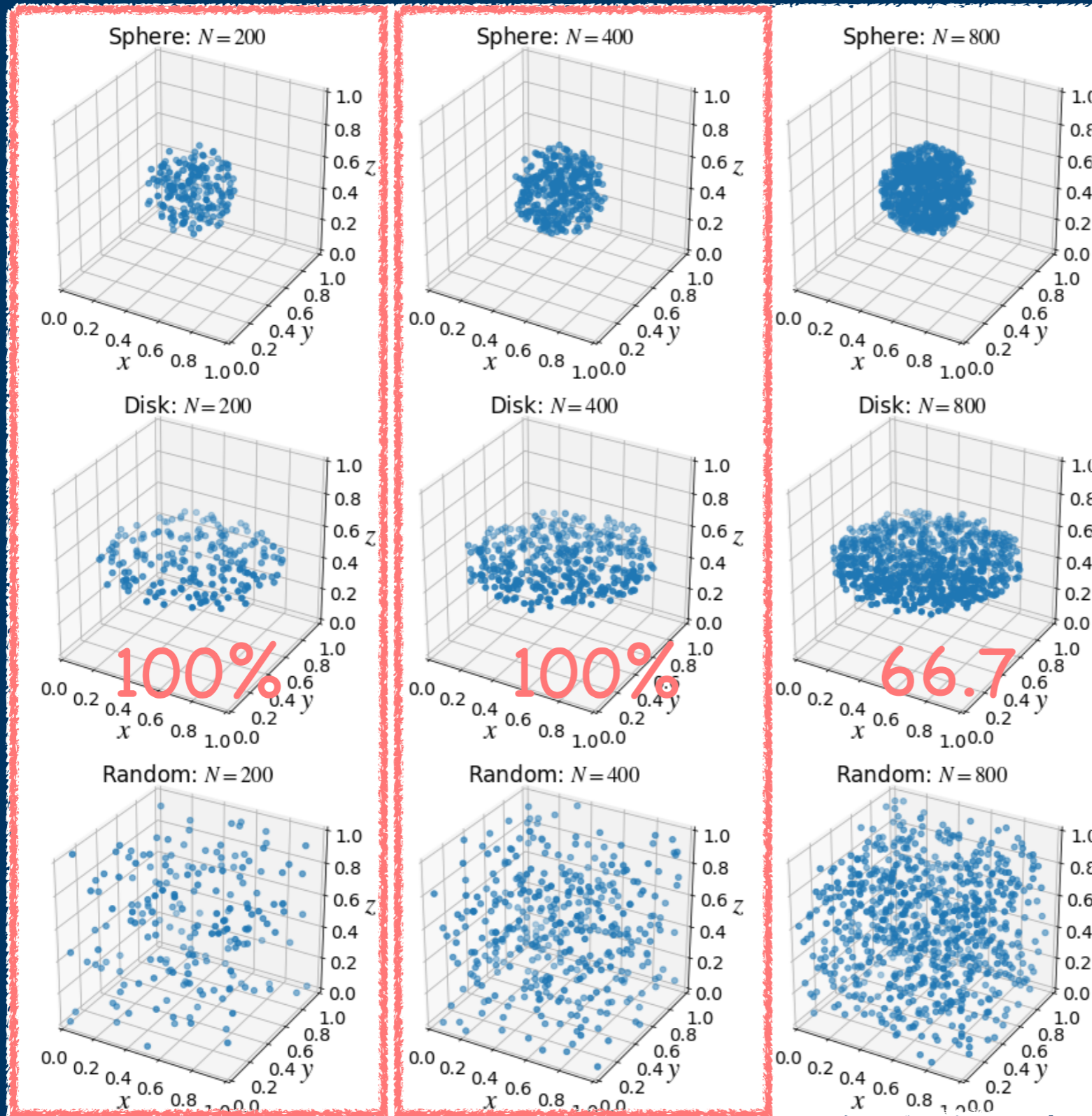
Example: Point Clouds and the Universe



Example: Point Clouds and the Universe



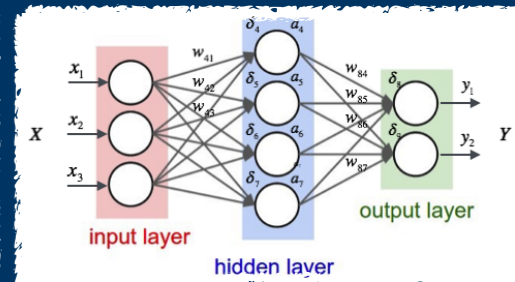
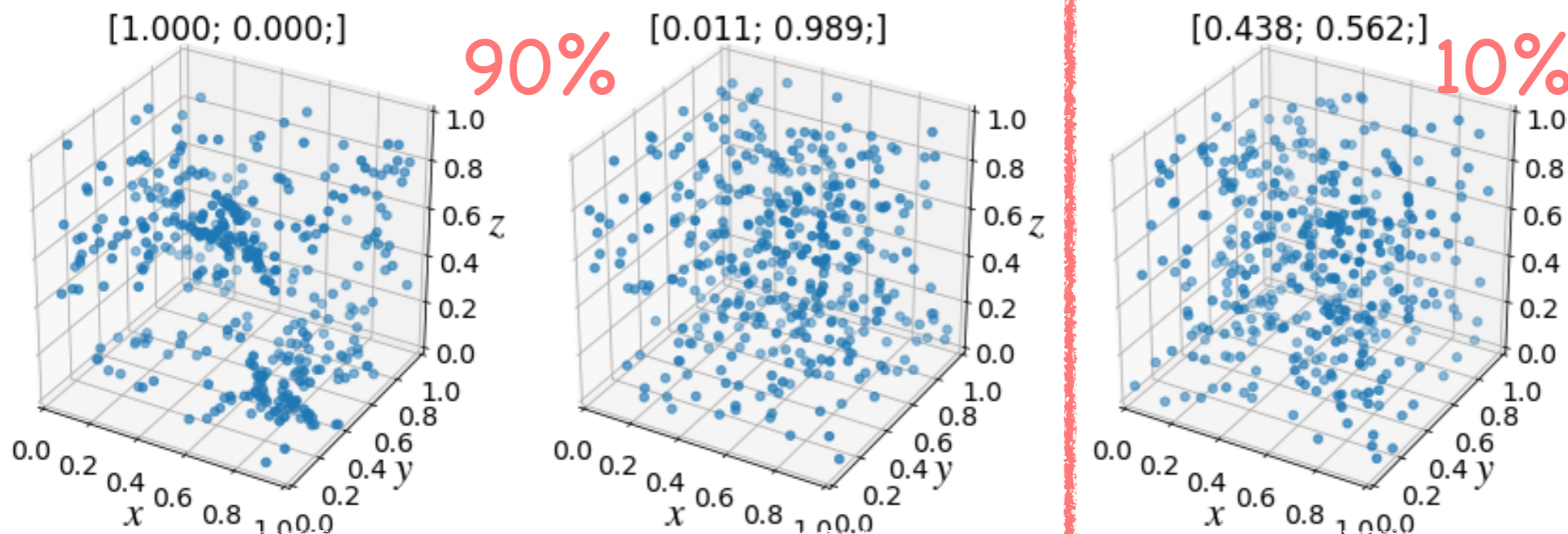
Example: Point Clouds and the Universe



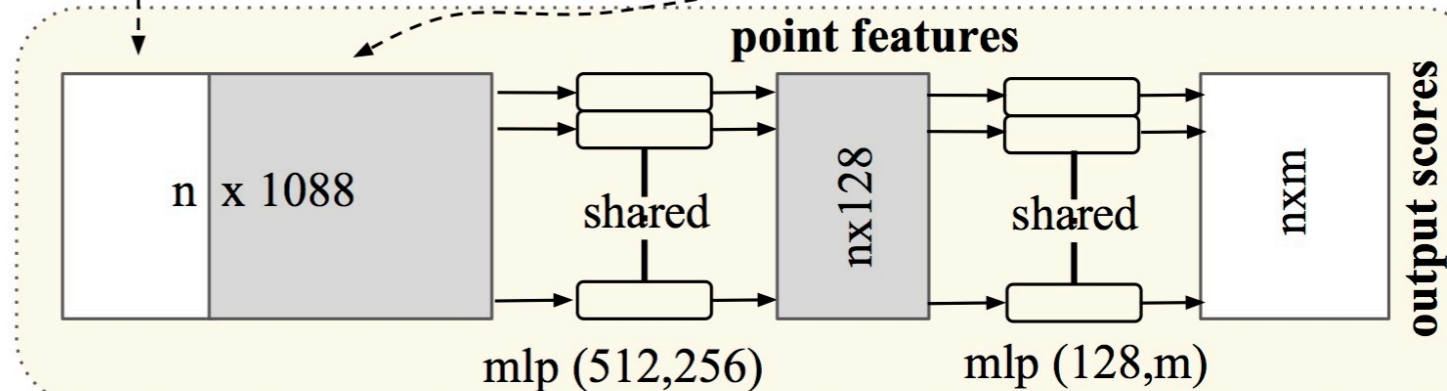
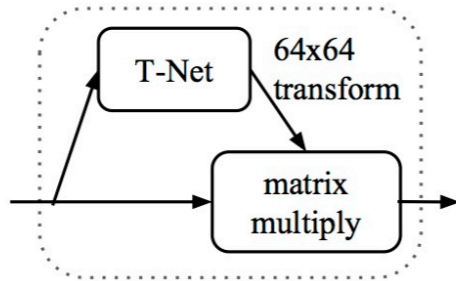
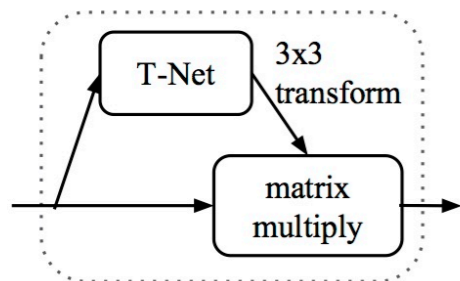
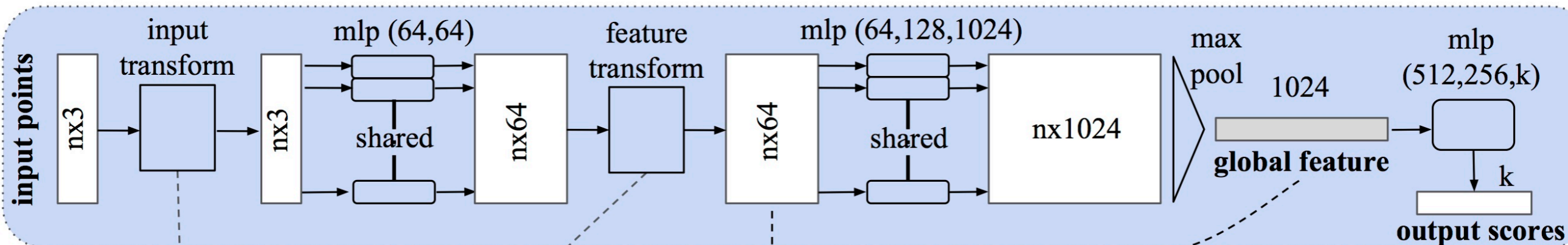
No...
Overfit !!



Example: Point Clouds and the Universe

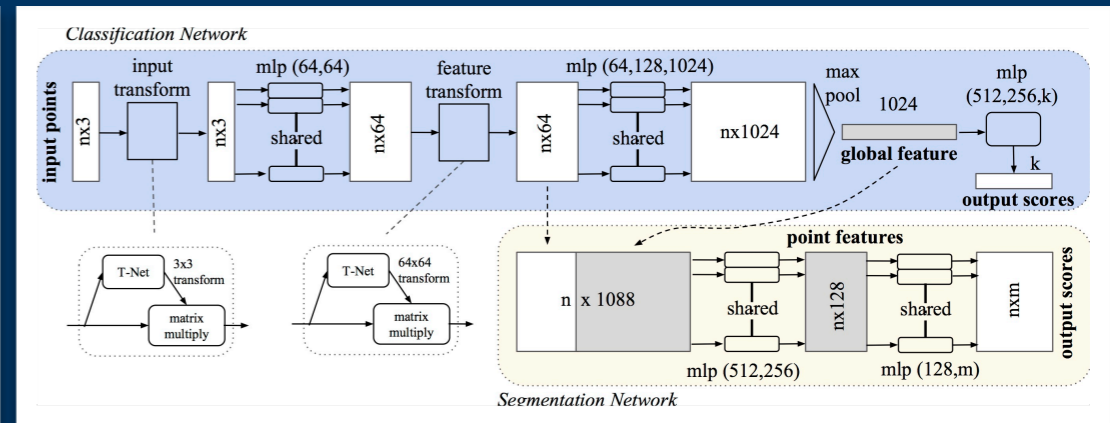
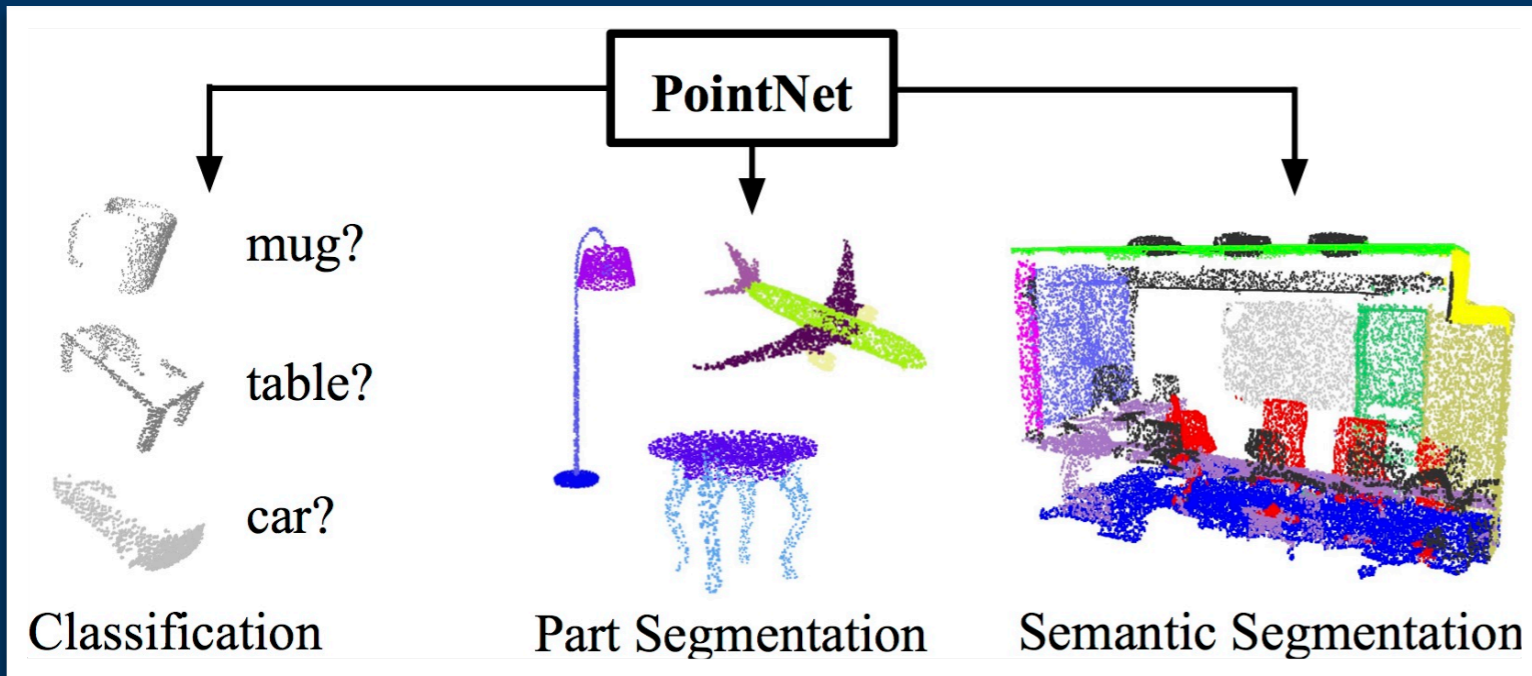
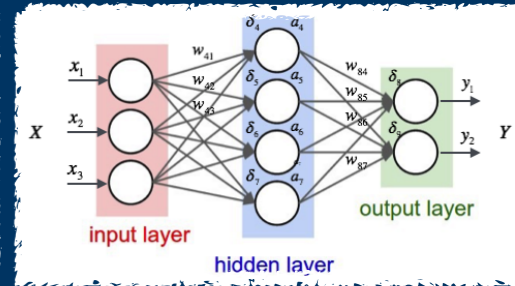
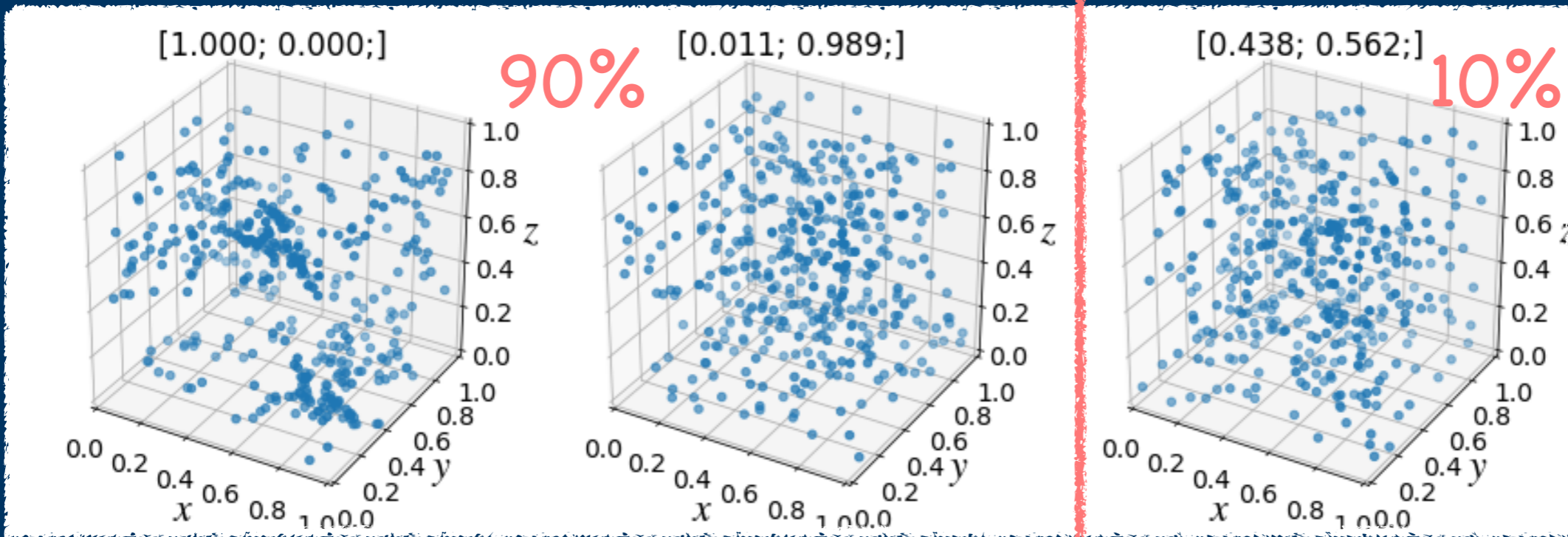


Classification Network



Segmentation Network

Example: Point Clouds and the Universe



Thank you!