

# **BigDataBench-MT: Multi-tenancy version of BigDataBench**

**Gang Lu**

**Beijing Academy of Frontier Science and Technology**

**BigDataBench Tutorial, ASPLOS 2016  
Atlanta, GA, USA**

# Multi-tenancy software

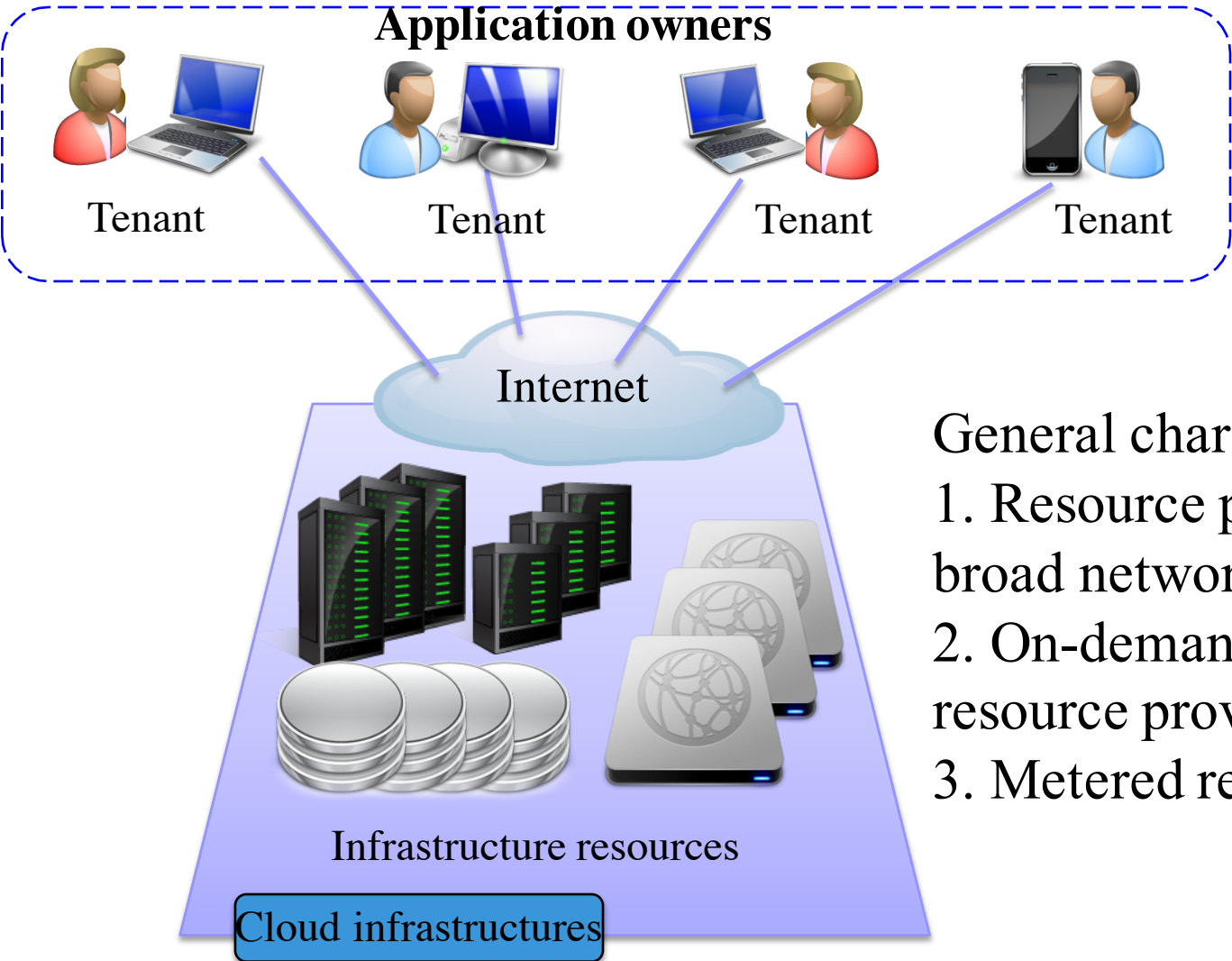
## ■ Software perspective

- Multi-tenancy refers to a principle in software architecture where **a single instance of the software runs on a server**, serving multiple client-organizations (tenants).
- With a multi-tenancy architecture, a software application is designed to ***virtually partition its data and configuration***, and each client organization works with a customized virtual application.

# Problems of single-tenancy benchmarks

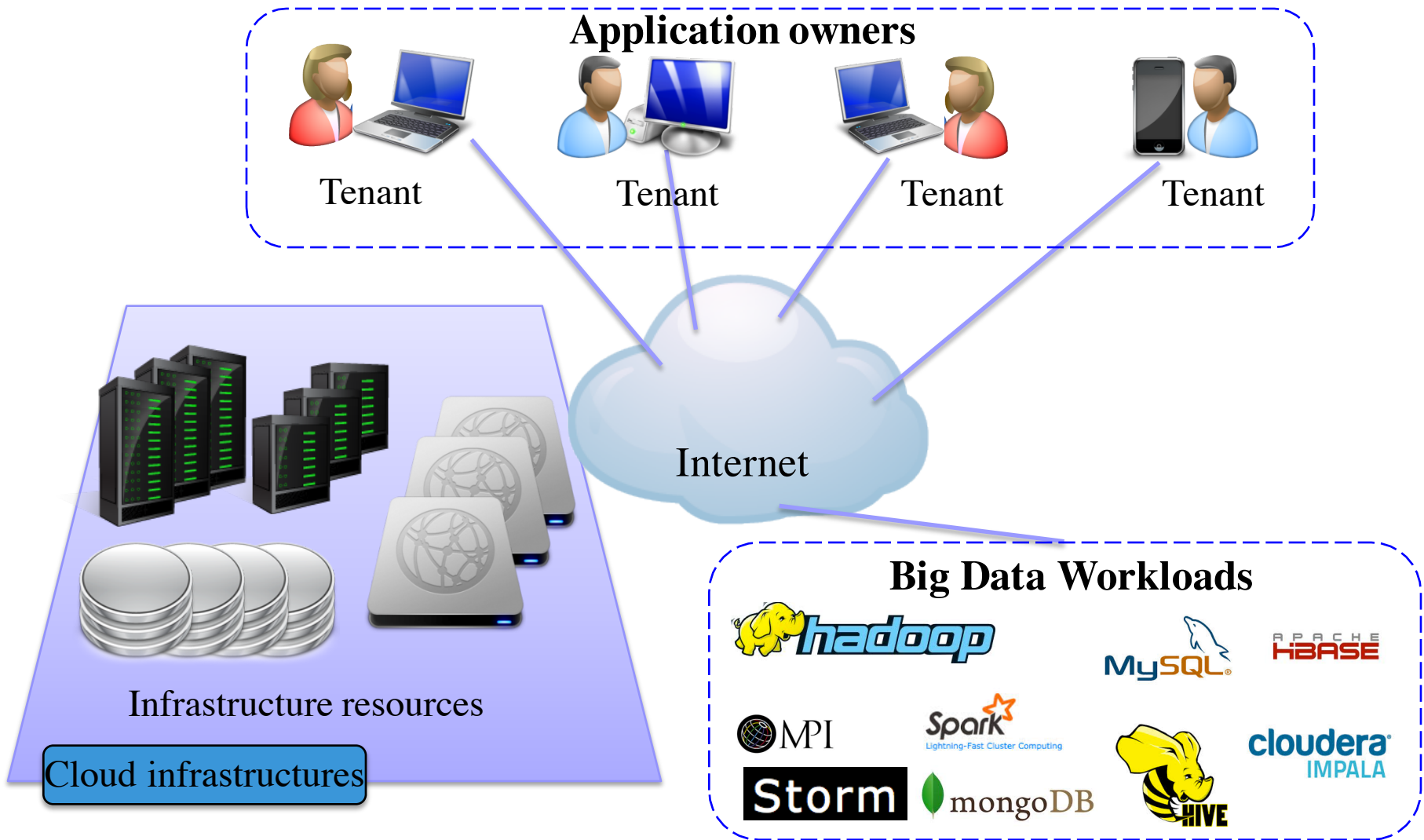
- Focus on **a single run** of workload
- Scenarios are not realistic (**simple and synthetic**)!
  - Does not match the typical operating conditions of real systems , in which *mixes of different percentages of tenants and workloads share the same computing infrastructure*
- We need to
  - Emulate real-world datacenter cluster with different amounts of tenants and various workload types and consequently various benchmarking scenarios.

# Multi-tenancy infrastructure

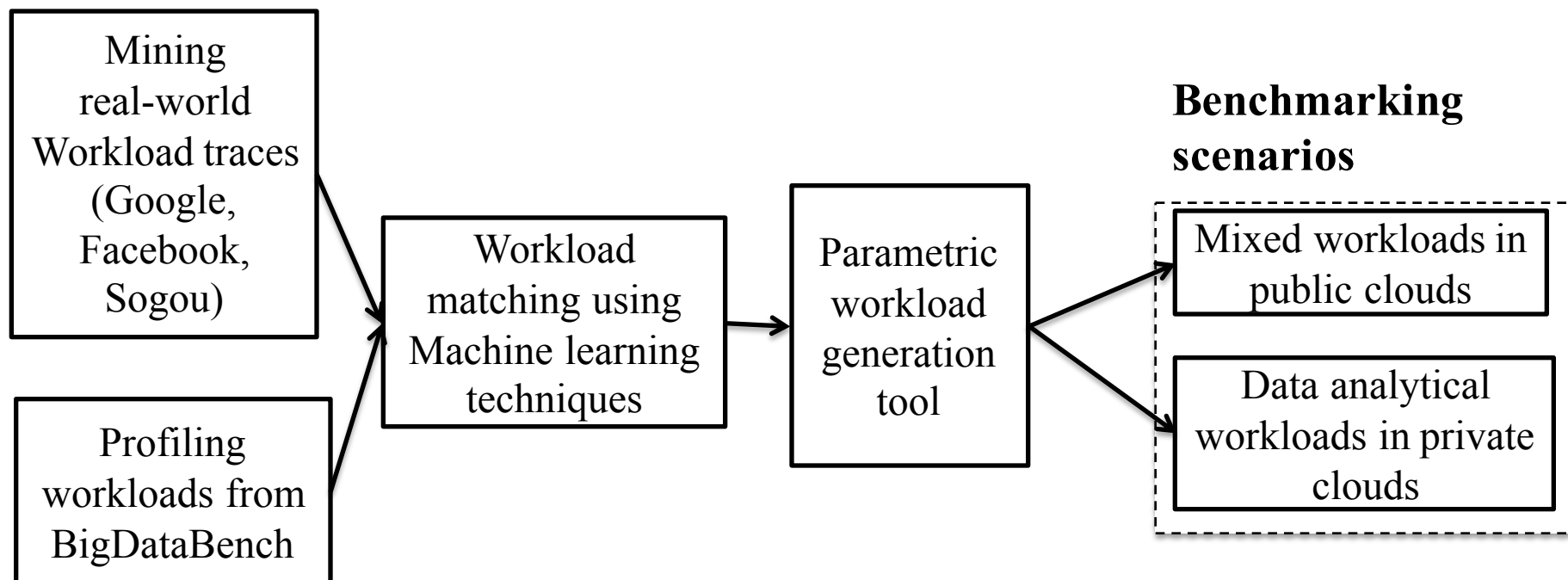


- General characteristics:
- 1. Resource pooling and broad network access
  - 2. On-demand and elastic resource provision
  - 3. Metered resources

# Multi-tenancy workloads



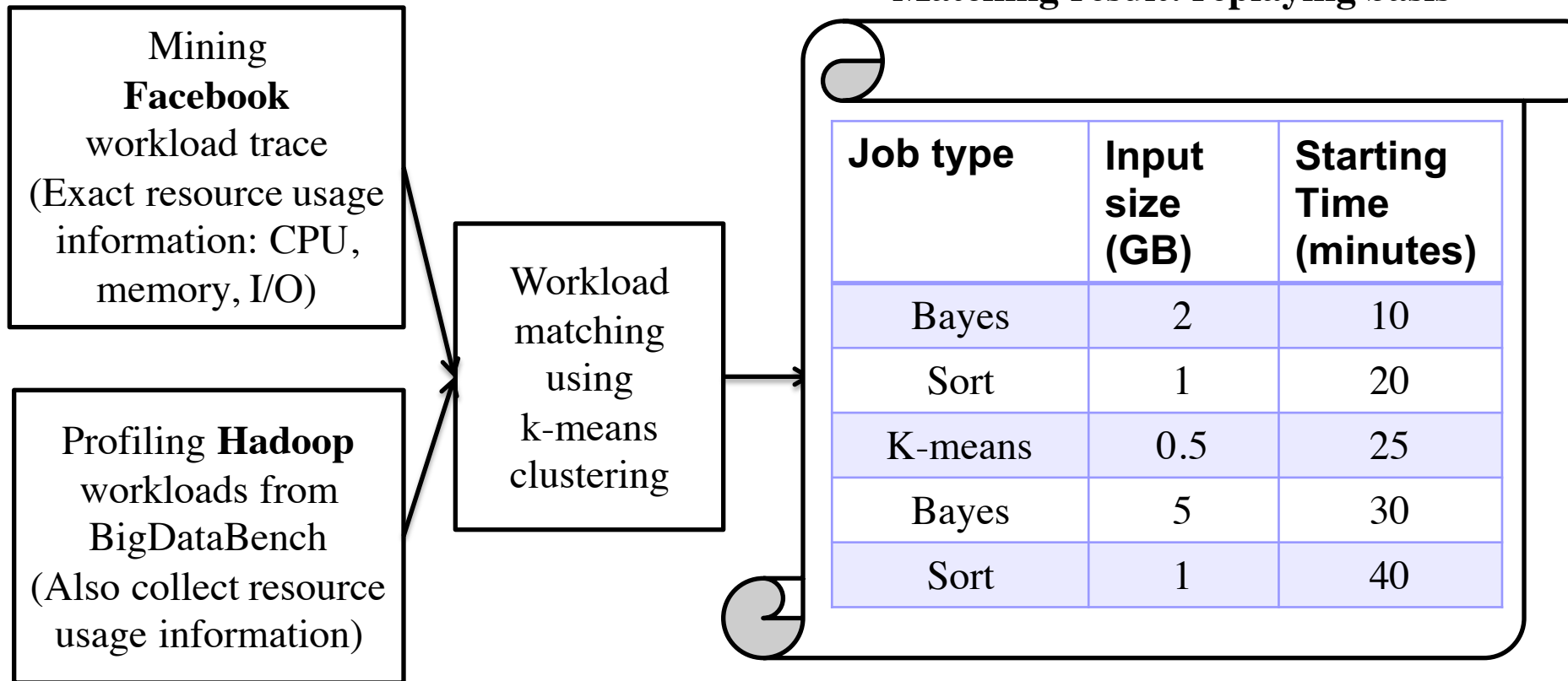
# BigDataBench-MT: Multi-tenancy version of BigDataBench



# Example: Hadoop workloads

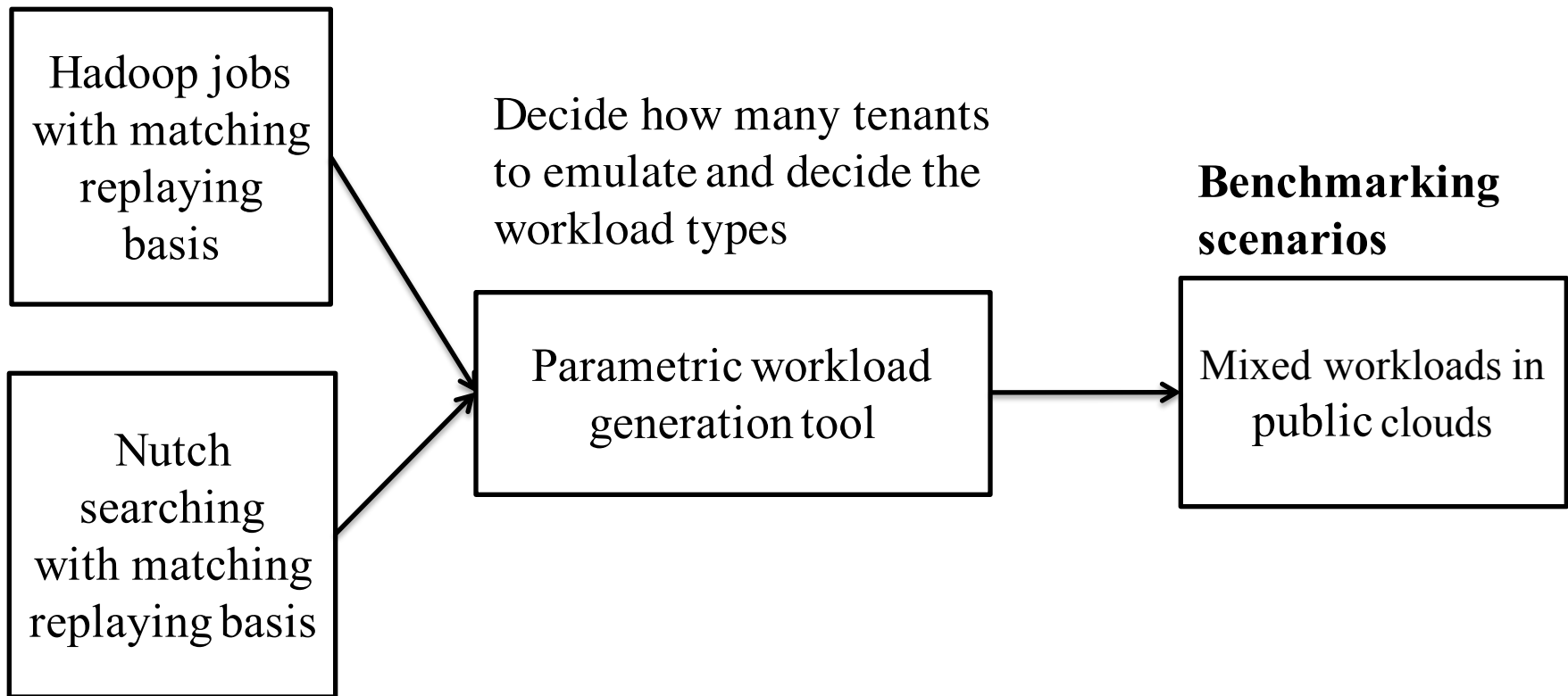
## ■ An example of match Hadoop workloads

Matching result: replaying basis



# Example: mixed workloads

## ■ An example of mixing search engine and Hadoop workloads



# What can you do with BigDataBench-MT?

- We consider two dimensions of the benchmarking scenarios
  - From tenants' perspectives
  - From workloads' perspectives

# You can specify the tenants

## ■ The number of tenants

- ☐ Scalability Benchmark: How many tenants are able to run in parallel ?

## ■ The priorities of tenants

- ☐ Fairness Benchmark: How fair is the system, i.e., are the available resources equally available to all tenants? If tenants have different priorities ?

## ■ Time line

- ☐ How the number and priorities of tenants change over time?

# You can specify the workloads

## ■ Data characteristics

- ☐ Data type, source
- ☐ Input/output data volumes, distributions

## ■ Computation semantics

- ☐ Algorithms
- ☐ Big data software stacks

## ■ Job arrival patterns

- ☐ Arrival rate
- ☐ Arrival sequence

# You can specify the interference

## ■ Each individual tenant:

- ☐ Different types of workloads
- ☐ How do they interfere each other at different resource dimensionalities?

## ■ Multiple tenants:

- ☐ How well are tenants isolated from one another with respect to performance?
- ☐ How do individual tenants influence other tenants' performance?

# Current status

## ■ Multi-tenancy V1.0 releases:

- Emulate workloads based on real-world workload traces
- Support mixes of both online service and offline batch workloads

Workloads	Software stack	Workload trace
Nutch Web Search	Apache Tomcat 6.0.26, Search Server (Nutch)	Sogou ( <a href="http://www.sogou.com/labs/dl/q-e.html">http://www.sogou.com/labs/dl/q-e.html</a> )
Hadoop	Hadoop 1.0.2	Facebook ( <a href="https://github.com/SWIMProjectUCB/SWIM/wiki">https://github.com/SWIMProjectUCB/SWIM/wiki</a> )
Shark	Shark 0.8.0	Google data center ( <a href="https://code.google.com/p/googleclusterdata/">https://code.google.com/p/googleclusterdata/</a> )

