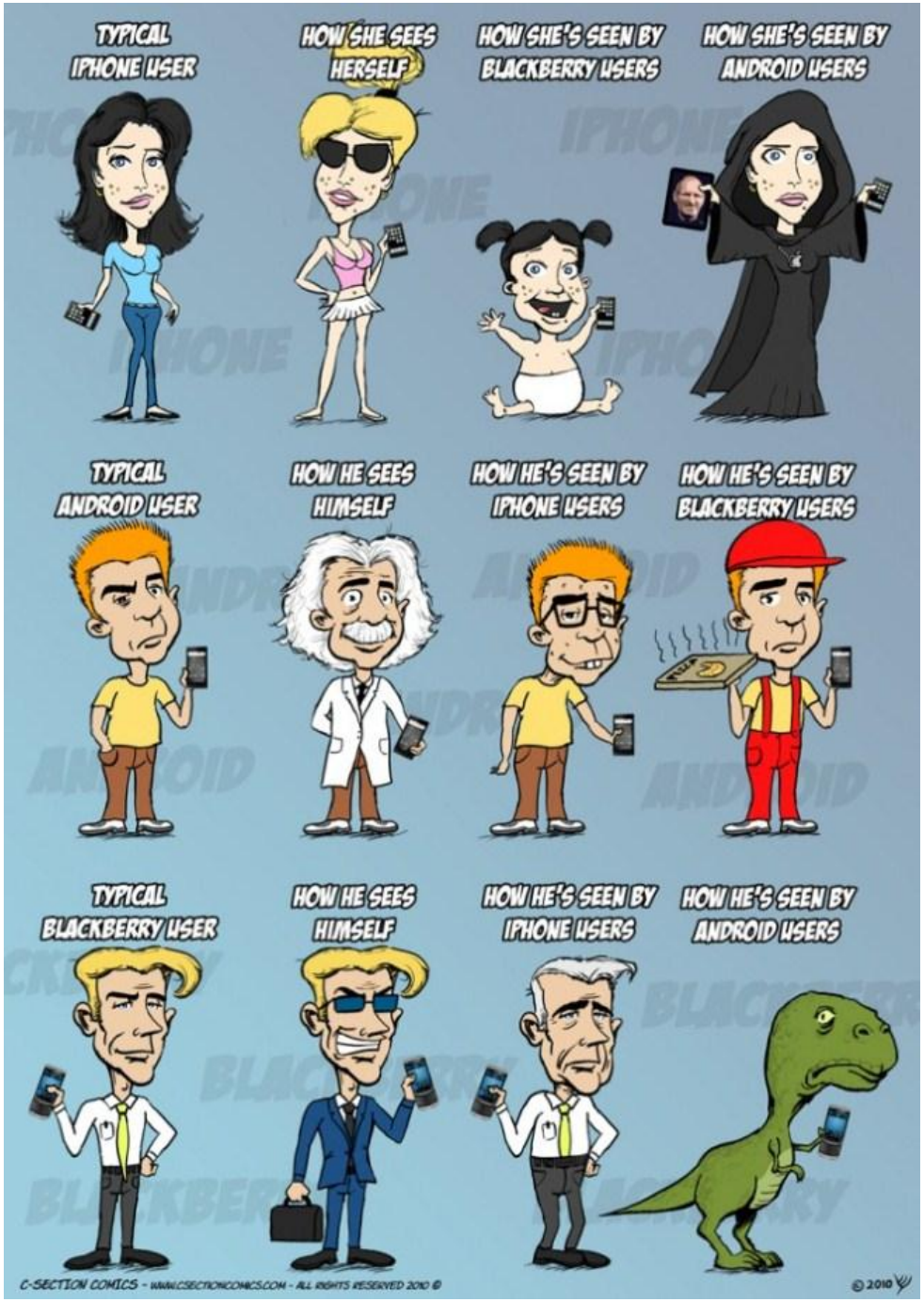
Toward to next-generation database

Yen-Liang Su
National Cheng Kung University, Tainan



PhiCloud
PRIVATE CLOUD PROVIDER

Who am I

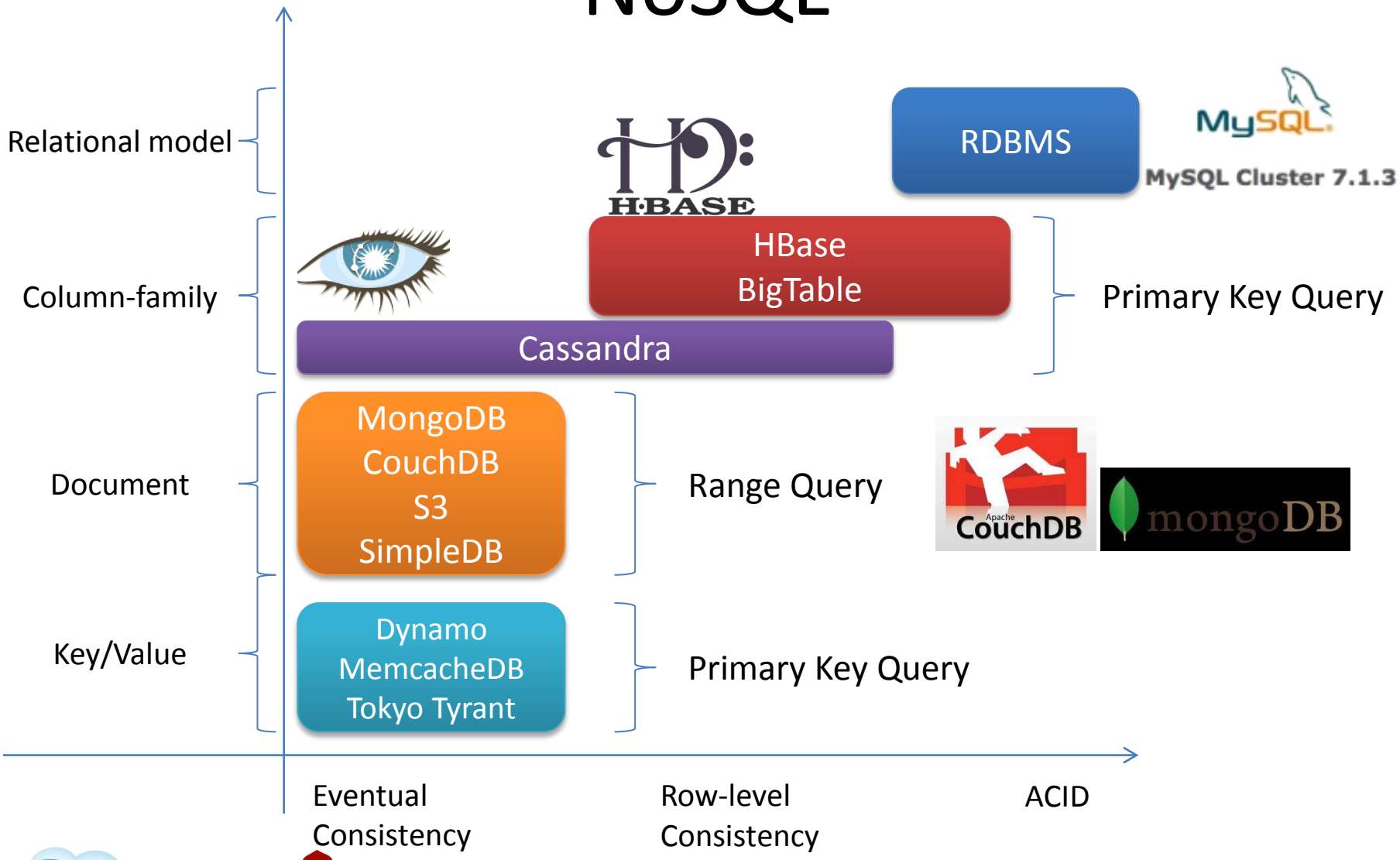
- A Master Student
  - HPDS Lab, Inst. of CCE, NCKU
  - 謝錫堃教授
- An Entrepreneur
  - PhiCloud.com

- Research Interest
  - Distributed File System
    - On-demand Data Co-allocation with User-Level Cache for Grids
      - Concurrency and Computation: Practice and Experience
  - MapReduce
    - Variable-Sized Map and Locality-Aware Reduce on Public-Resource Grids
      - GPC 2010
  - Distributed Transaction Processing
    - TSorter: A Conflict-Aware Transaction Processing System for Clouds
      - NCKU master thesis

Outline

- NoSQL
- Existing Transactional NoSQL
- TSorter
- Hadoop/HBase Experience
- Future Work

NoSQL


Relational model
Column-family
Document
Key/Value
RDBMS
MySQL Cluster 7.1.3
HBase
BigTable
Cassandra
Primary Key Query
MongoDB
CouchDB
S3
SimpleDB
Range Query
Dynamo
MemcacheDB
Tokyo Tyrant
Primary Key Query
Eventual Consistency
Row-level Consistency
ACID

Characteristics

- **Schema-Free**
- **Easy Replication Support**
- **Simple API**
- **Eventually Consistent**
- **BASE**
  - Basically Available, Soft State, Eventually Consistency
- **Huge Data Amount**

From http://nosql-database.org/

Database Usage

- Analytical Processing
  - MapReduce Programming
  - Hive and Pig
- Transaction Processing
  - Lack support of ACID

ACID

- Atomicity
  - Group multiple operations
- Consistency
  - ~~Sequential/Eventual Consistency~~
  - Schema (However, NoSQL is schema-free)
- Isolation
  - Transaction serializability
- Durability
  - HDFS-200

Transactional HBase

- Optimistic Concurrency Control (OCC)
  - No Lock
  - So, Good Luck
- 2-Phase Commit (2PC)
  - Distributed Transaction Processing

Transactional HBase

T_1 :

value = 1

value = 2

value = get(row id=123, column="value")

value = value + 1

success

put(key

success

commit()

T_2 :

value = 1

value = 2

value = get(row id=123, column="value")

value = value + 1

success

put(key

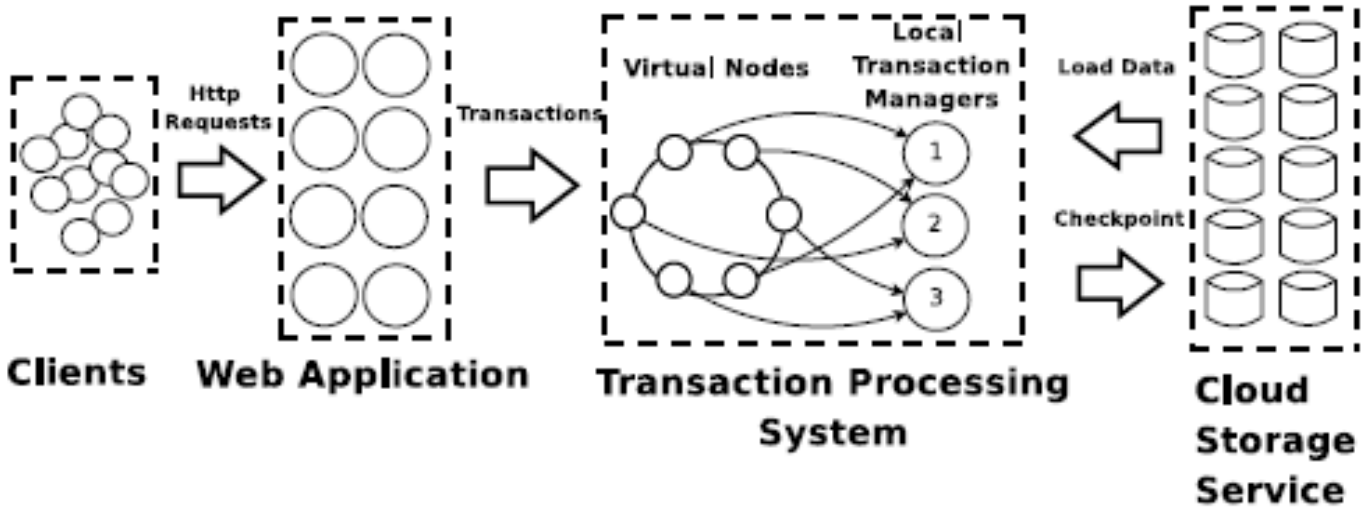
fail

commit()

Row id	Values
…	…
123	**{"value": 2}**
…	…

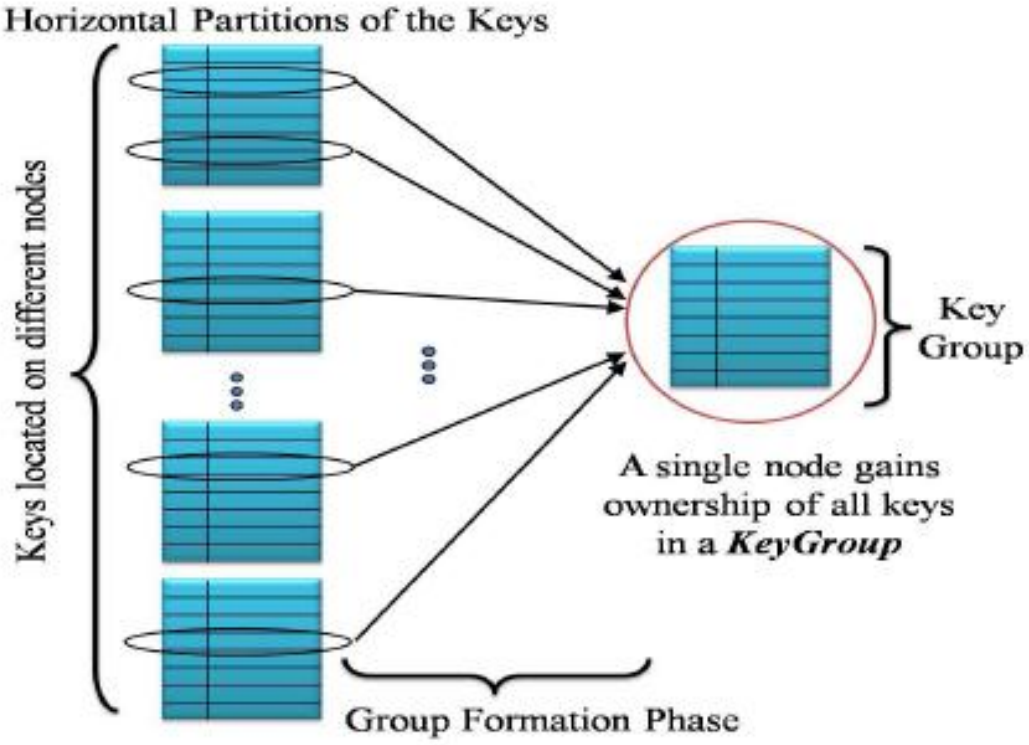
CloudTPS

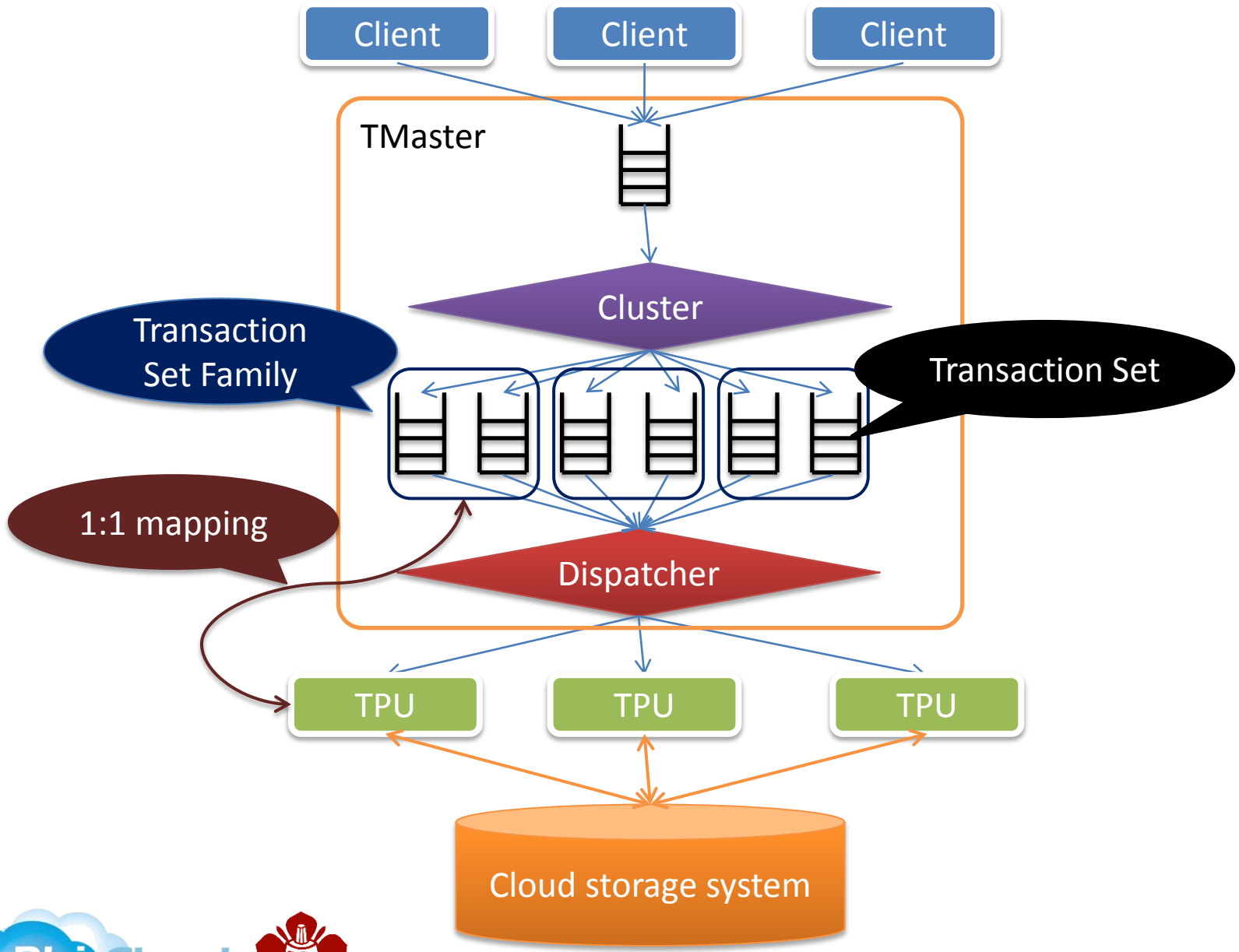
- Similar to Transactional HBase
- Timestamp-Based Concurrency Control
  - Sort sub-transactions by the global timestamp



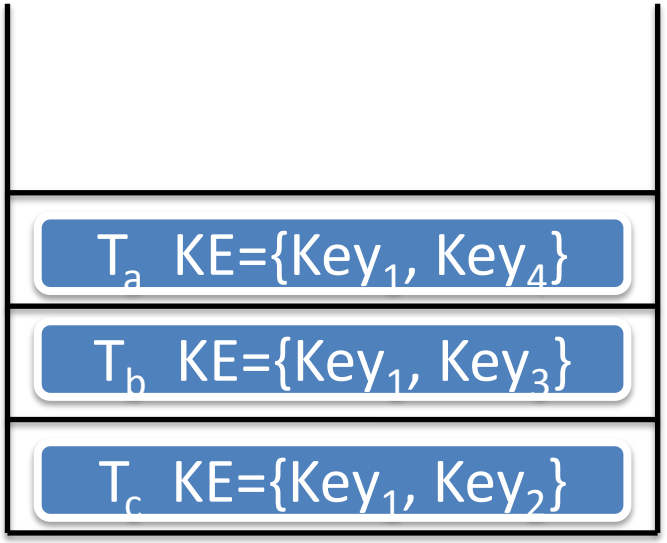
G-Store

- Key Grouping Protocol




Client
Client
Client
TMaster
Cluster
Transaction Set Family
Transaction Set
1:1 mapping
Dispatcher
TPU
TPU
TPU
Cloud storage system

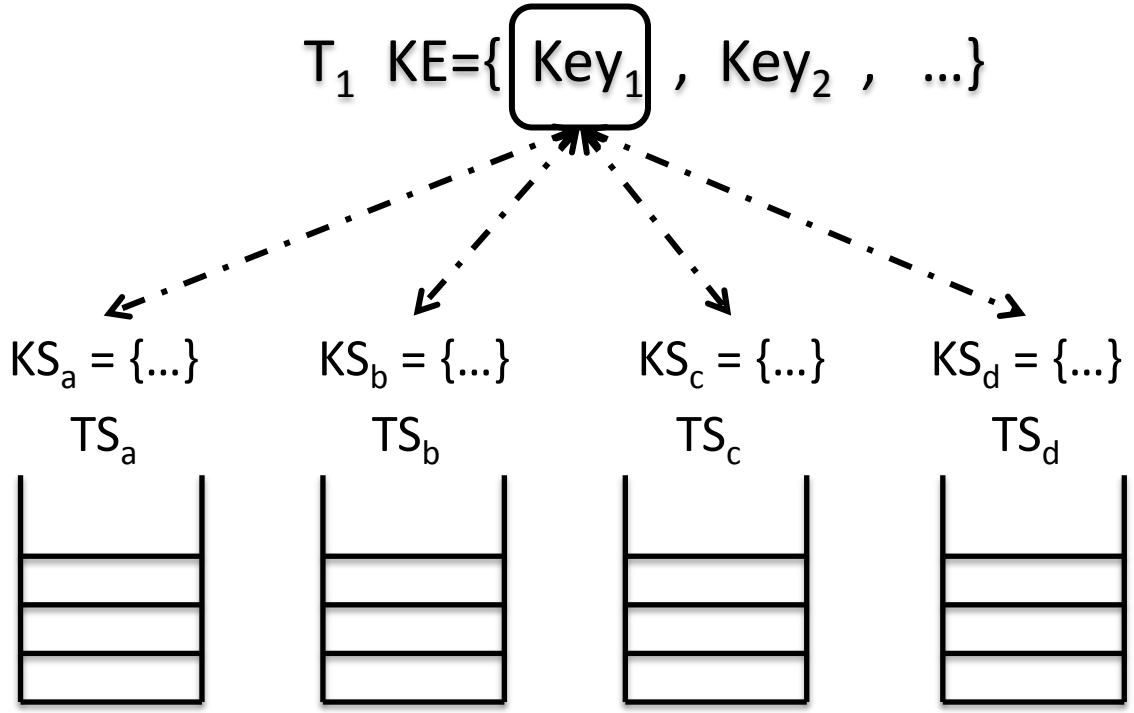
Key Set



TS_a

$KS_a = \{Key_1, Key_2, Key_3, Key_4\}$

Conflict Detection



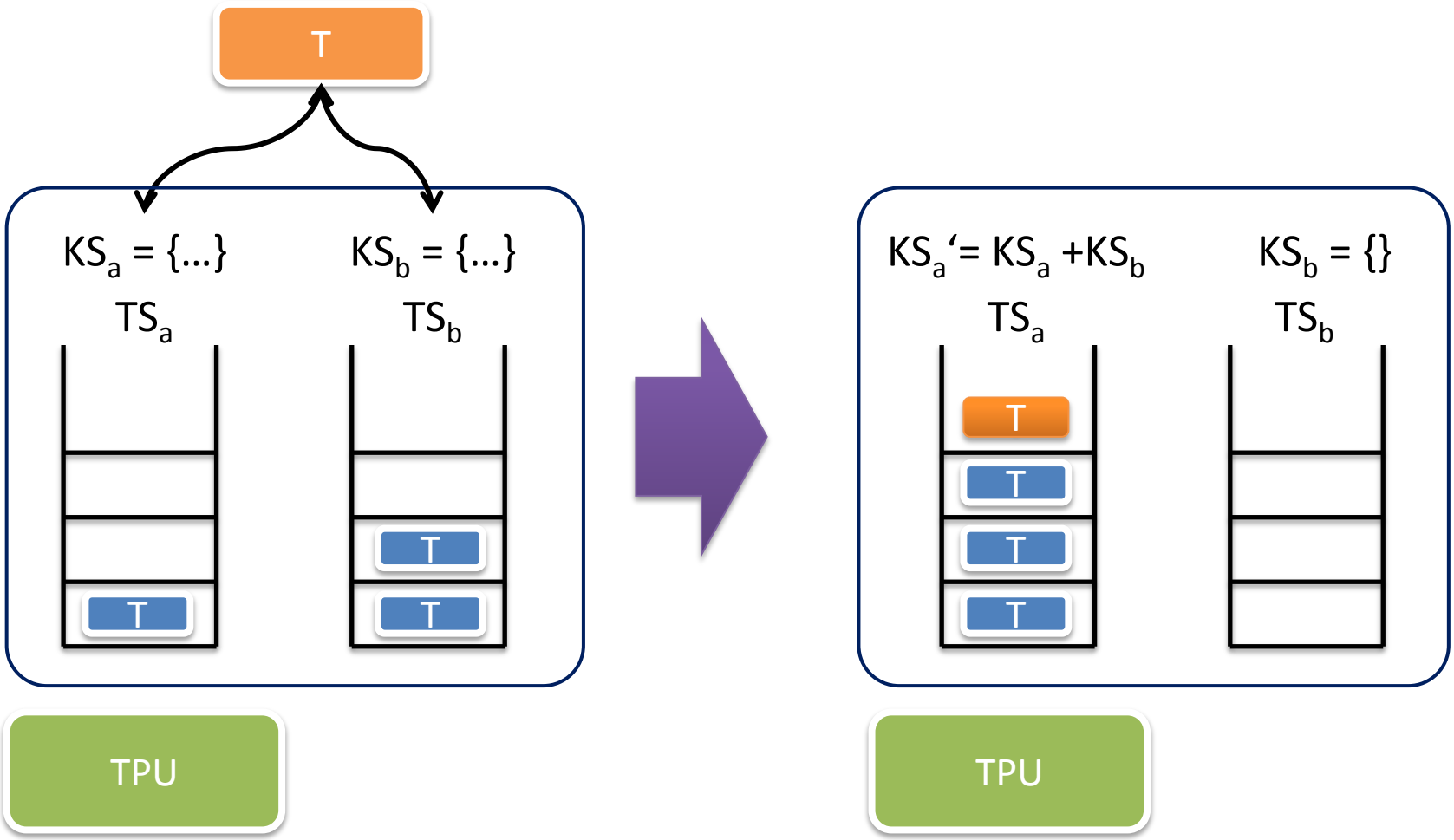
O(*the number of keys* × *the number of transaction sets*)

Conflict Detection

Key	Transaction set
Key_1	TS_a
Key_2	TS_b
Key_3	TS_c
Key_4	TS_d
…	…

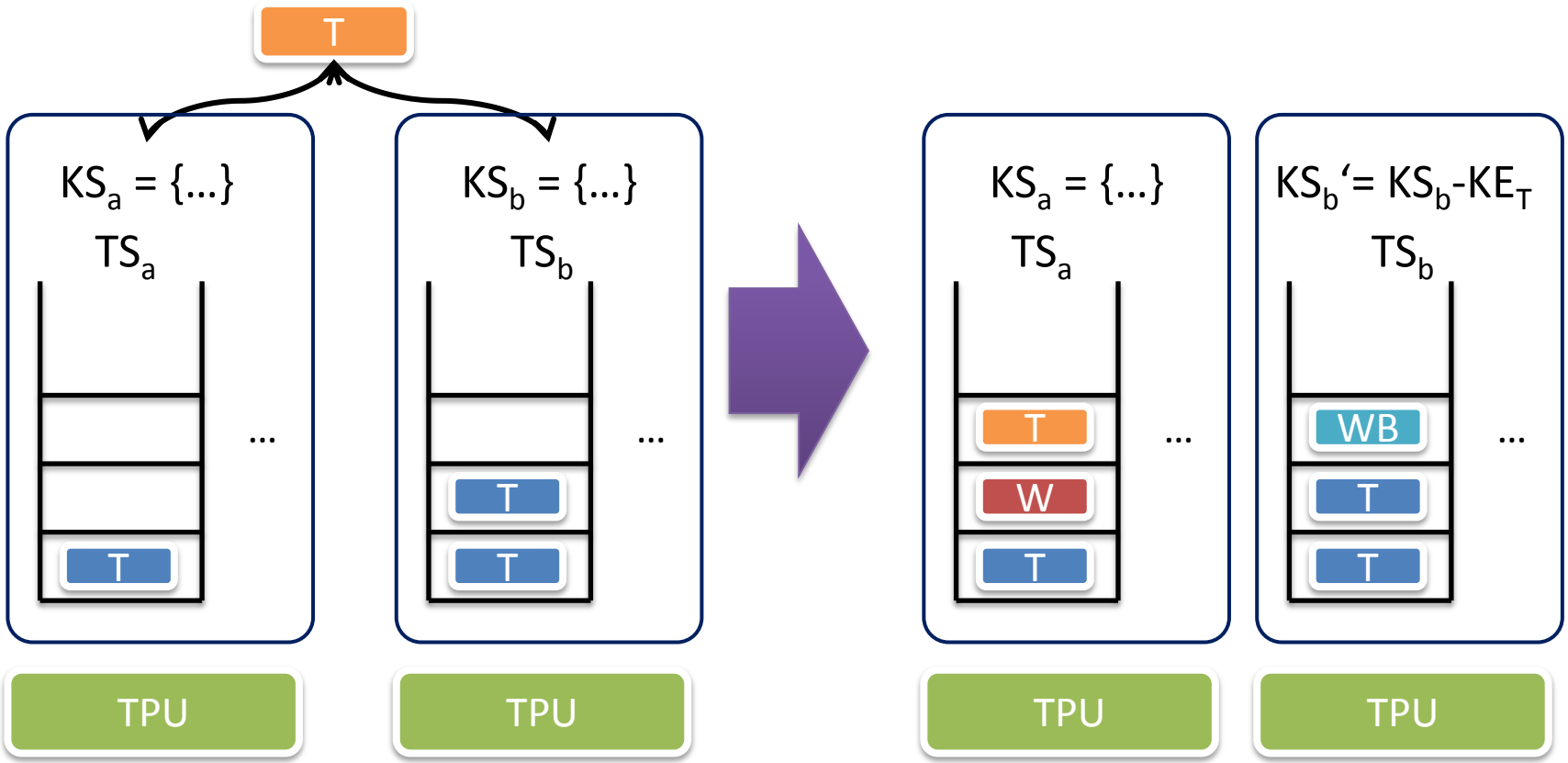
O(*the number of keys*)

Transaction Set Serialization

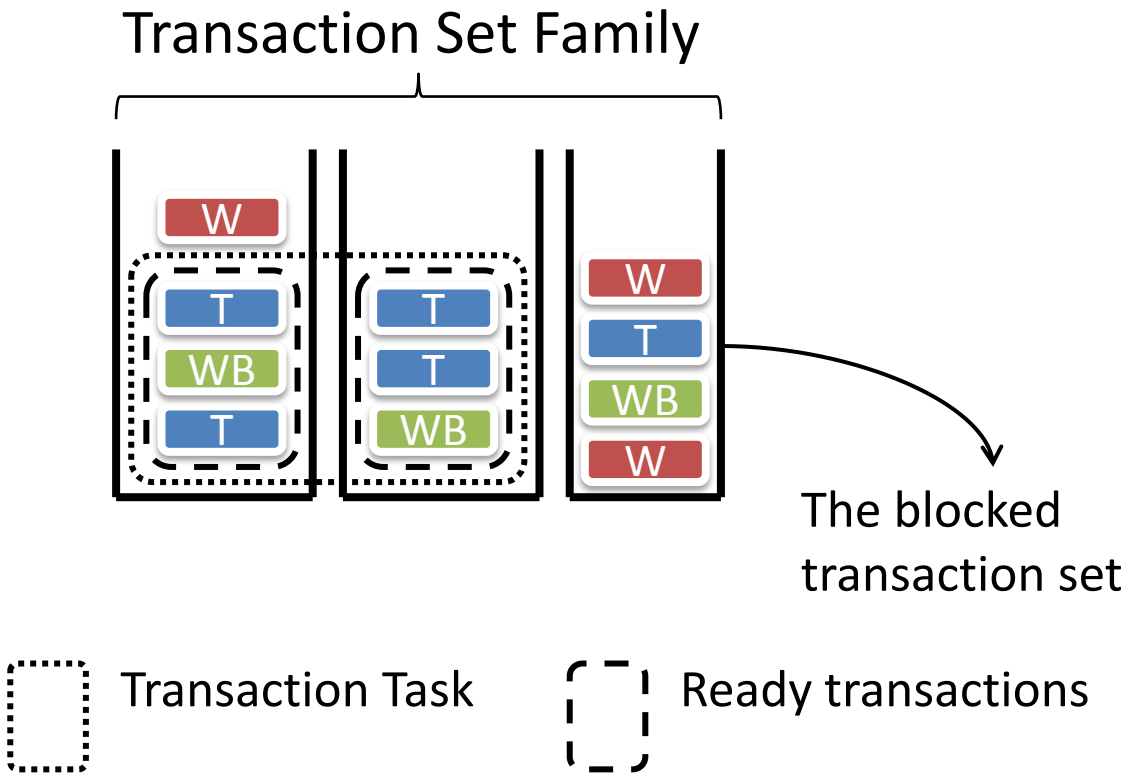


Transaction Set Serialization

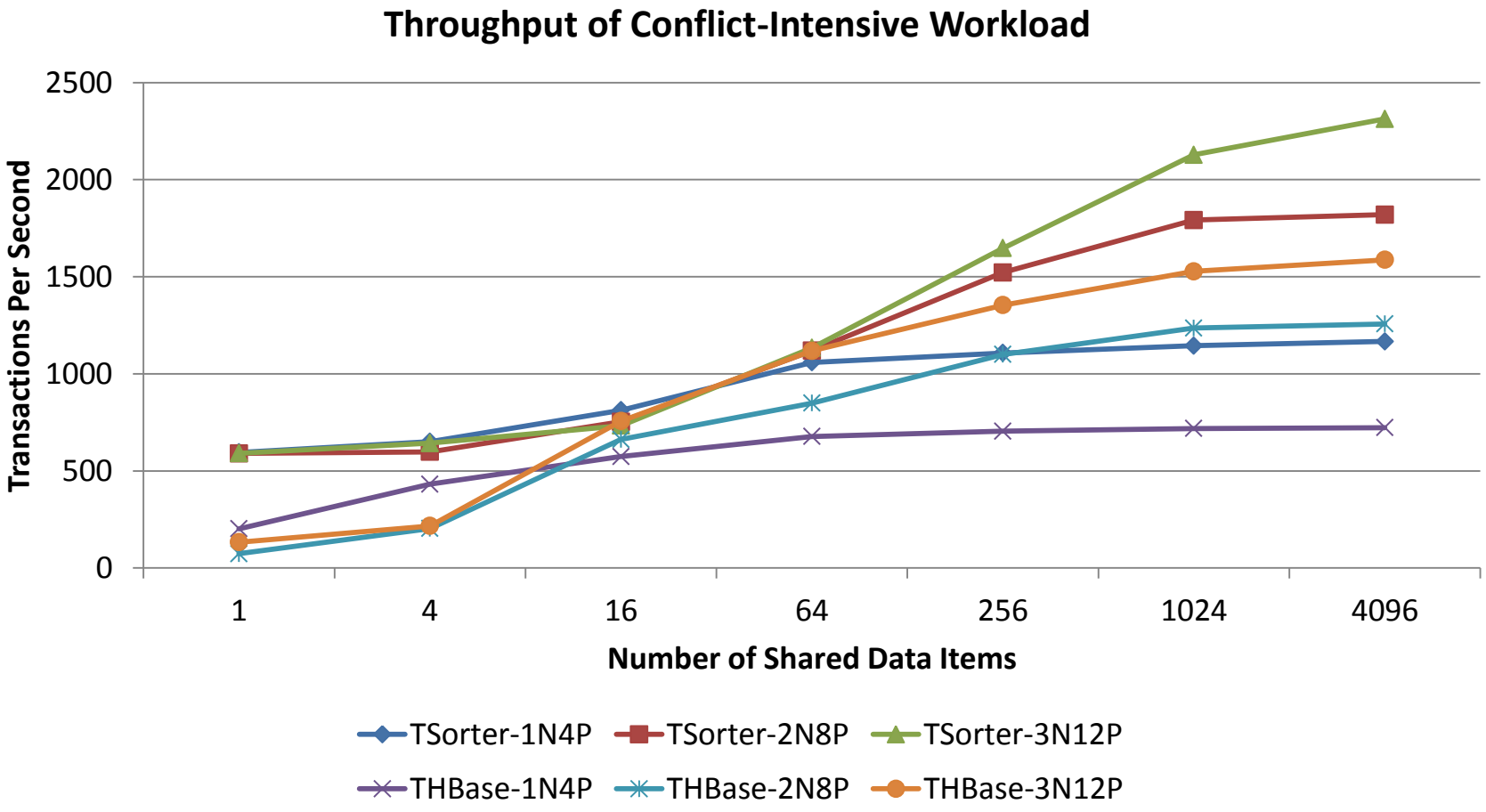
- Write-back transaction
- Wait transaction



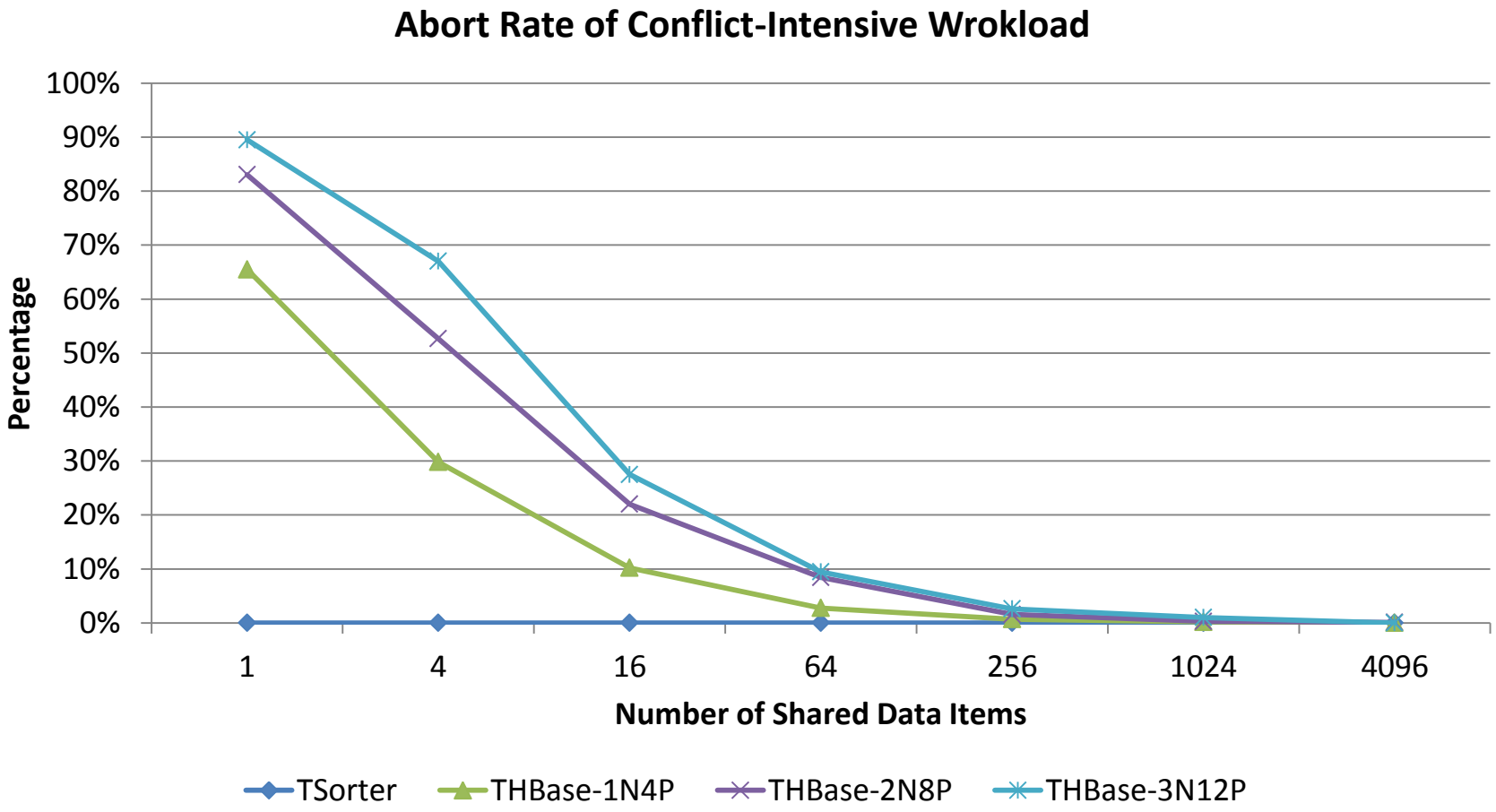
Transaction Dispatcher



Experiment


Throughput of Conflict-Intensive Workload
Transactions Per Second
2500
2000
1500
1000
500
0
1
4
16
64
256
1024
4096
Number of Shared Data Items
TSorter-1N4P
TSorter-2N8P
TSorter-3N12P
THBase-1N4P
THBase-2N8P
THBase-3N12P

Experiment


Abort Rate of Conflict-Intensive Wrokload
Percentage
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1
4
16
64
256
1024
4096
Number of Shared Data Items
TSorter
THBase-1N4P
THBase-2N8P
THBase-3N12P

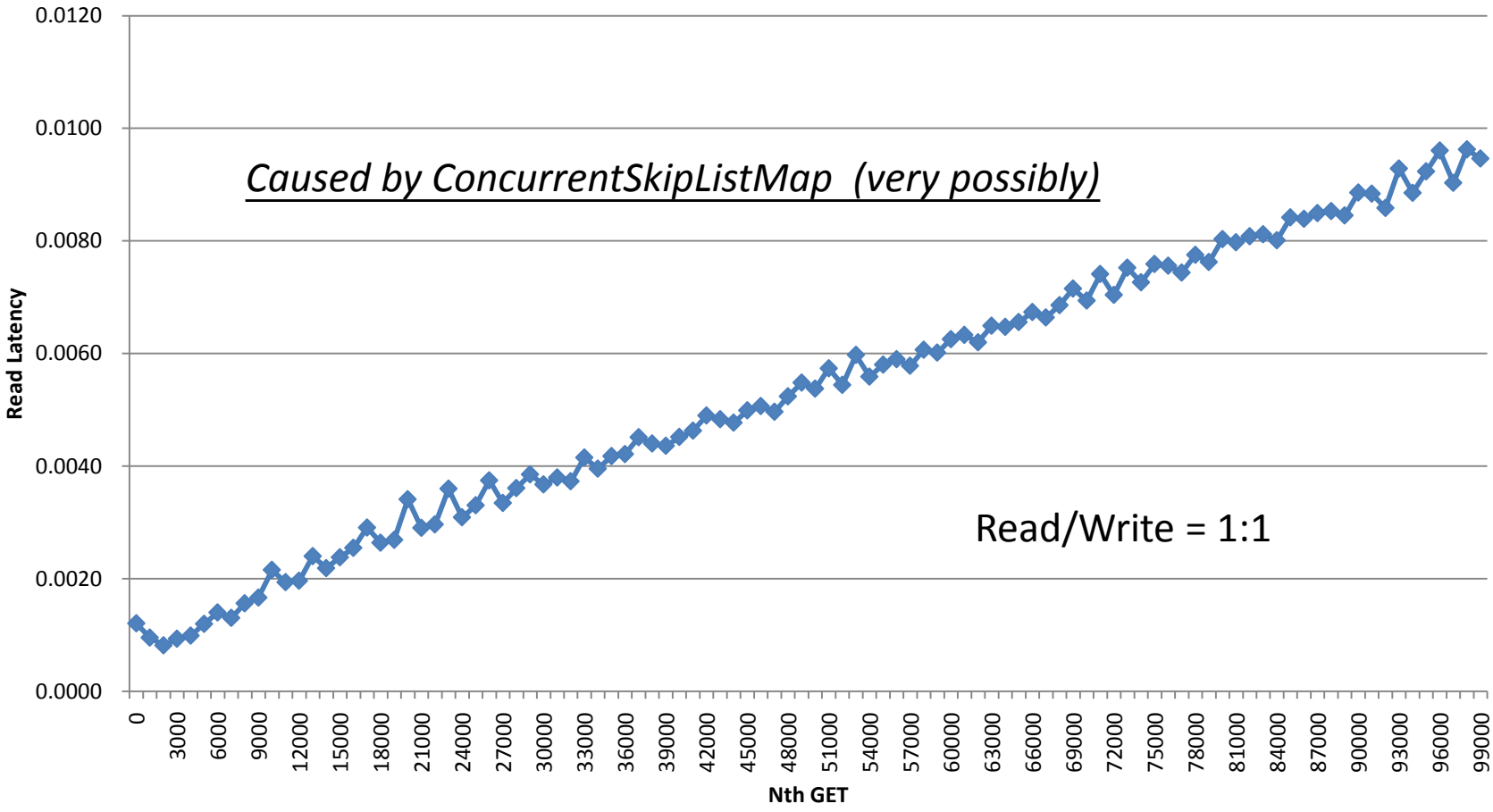
Hadoop/HBase Experience

- Hadoop
  - Q1: HDFS as a Generic File System
- HBase
  - Q2: High Latency Read of HBase
  - Q3: HBase as a Web-Backend
  - Q4: HBase vs Cassandra

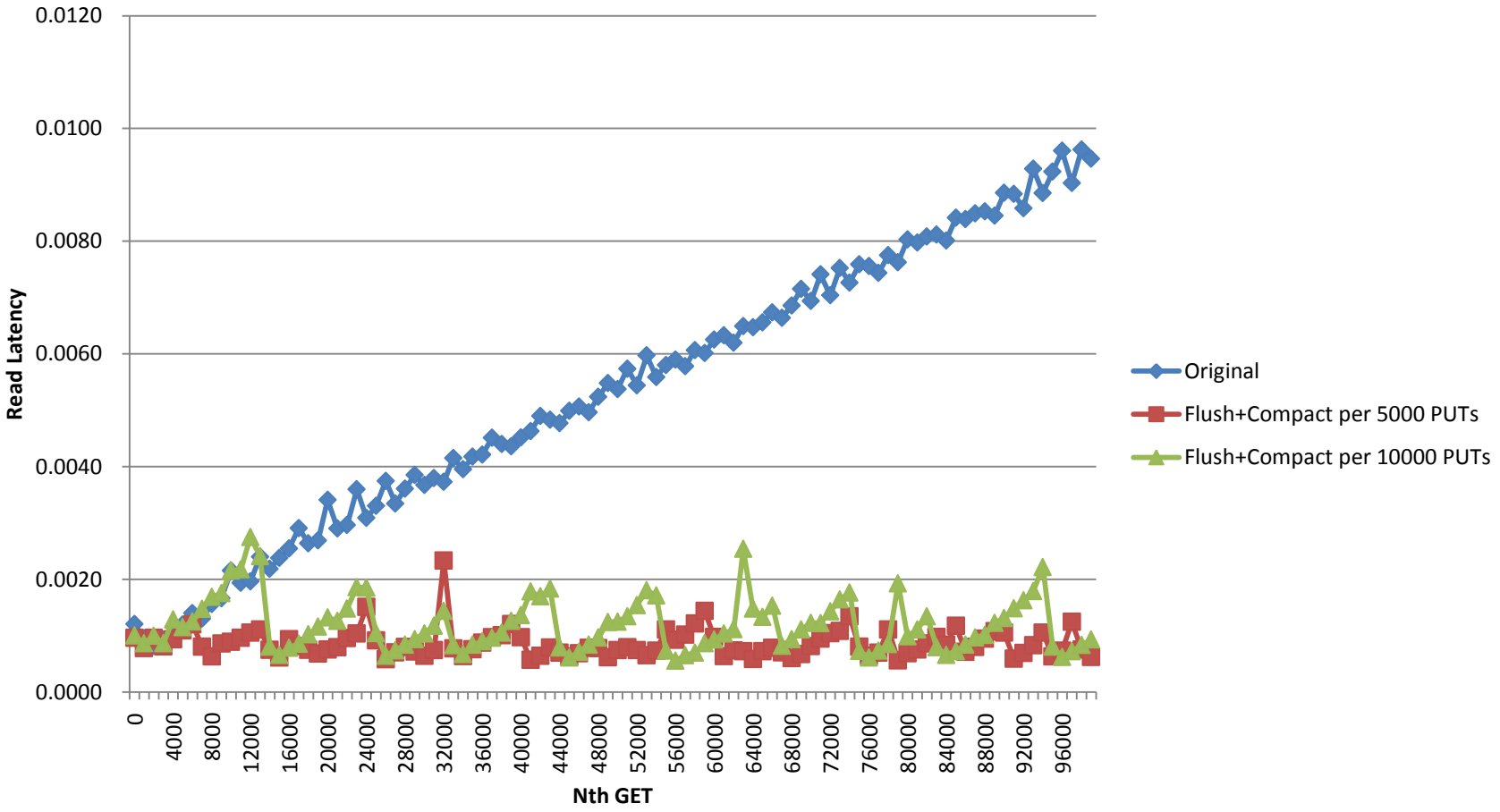
HDFS as a Generic File System

- HDFS is not a POSIX-compliant FS
  - Write-Once-Read-Many
- You can Try GlusterFS
  - GlusterFS is one of the most sophisticated file system in terms of features and extensibility.
  - Features
    - Data Replicated/Striped
    - No metadata
    - Client-Side Implementation

High Latency Read of HBase



High Latency Read of HBase



HBase as a Web-Backend

- ***wbbs.cc.ncku.edu.tw***
  - ~750,000 Rows
  - ~1.5 GB
    - ~50 MB Index
  - ~3000 requests per day
  - I use MySQL
- *力可科技*
  - ~ 1 Billion Rows
  - ~55 GB
    - ~24 GB Index
  - They use Cassandra

- Size
  - GB? TB? PB?
- Requests per Second
- Database Consolidation
  - Very Large Migration Cost

HBase vs Cassandra

	HBase	Cassandra
Storage Layer	HDFS	No
Data Partition	Size-Based	Random
Data Cache	Block-Based (1GB=8192 Blocks)	Row-Based
Consistency	Row-Level Serialization	Eventual Consistency • Write-One-Read-All • Write-All-Read-One • (W+R) > (N/2)

Suitable for	• Analytical Processing
	  • Search Engine
	• OLTP with the Strong Need of Serialization
	  • FB new IM System
		OLTP
		• Mail
		• Log
		• Content Repo.

Scalability ?

~~Big and Large~~

From Small To Large

PhiCloud

- **PhiCenter**
  - 2011 Q1
  - A System Like Eucalyptus, OpenNebula, OpenStack and XenServer
  - Easy-to-Use, Based-on VirtualBox-OSE and GPLed

PhiCloud

- **PhiDesktop**
  - 2011 Q2
  - A Product Like VMWare View or XenDesktop
  - Address on University
    - 校園授權軟體
  - 50% Features and 20% Costs
    - V牌 USD150
    - C牌 USD100
  - Components
    - VirtualBox-OSE as Hypervisor, PhiCenter as VM Management
    - Xvfb as X11 Enviroment
    - XDamage Extension for Damage Detection
    - libjpeg-turbo for Image Compression

PhiCloud

- **PhiData**
  - 2011 Q2
  - Public Cloud of Data Warehouse as a Service
  - Hive-Based or Develop a new MapReduce Engine
  - Welcome to Join PhiCloud