Yun-xiang Zhao, Wan-xin Zhang, Dong-sheng Li, Zhen Huang, Min-ne Li, Xi-cheng Lu, 2016. Pegasus: a distributed and load-balancing fingerprint identification system. *Frontiers of Information Technology & Electronic Engineering*, **17**(8):766-780. http://dx.doi.org/10.1631/FITEE.1500487

Pegasus: a distributed and load-balancing fingerprint identification system

Key words: Distributed fingerprint identification, Distributed MongoDB, Load balancing

Corresponding author: Dong-sheng Li
E-mail: dsli@nudt.edu.cn
ORCID: http://orcid.org/0000-0001-9743-2034

Motivation

- Fingerprint identification is one of the most widely used methods in the field of biometric authentication, thanks to its uniqueness, stability, and easy-to-collect.
- Existing fingerprint identification systems and related research have focused on parallelizing the process of preprocessing or feature extraction (Indrawan *et al*., 2011), which increases single fingerprint's processing speed substantially. However, due to the scenarios of massive users and concurrent access with the popularity of fingerprint in biometric identification systems and the coming of big data age, it is essential to take distributed or cluster computing into consideration.

Main idea

- Inspired by the observations that:
  - It is costly to use multiple nodes for parallelizing the feature extraction phase among small but massive files.
  - Existing databases consider only the volume of data among nodes to balance data storage without paying attention to the pressure of data accessing like query and insertion, which may cause breakdowns of nodes under high accessing pressure.

Method

1. Implement the feature extraction of fingerprints with the HIPI library to degrade the time consumption of processing massive fingerprints.
2. Use load-balancing distributed MongoDB which includes:
   - Using consistent hashing to balance access requests on front-end mongos nodes.
   - Optimizing distributed MongoDB's default load-balancing algorithm from the view of data operation of each shard.
3. Carry out various experiments based on the proposed system.

Major results

- Our system can deal with massive fingerprints faster than Hadoop.
- The front-end mongos are more load-balanced than default distributed MongoDB.

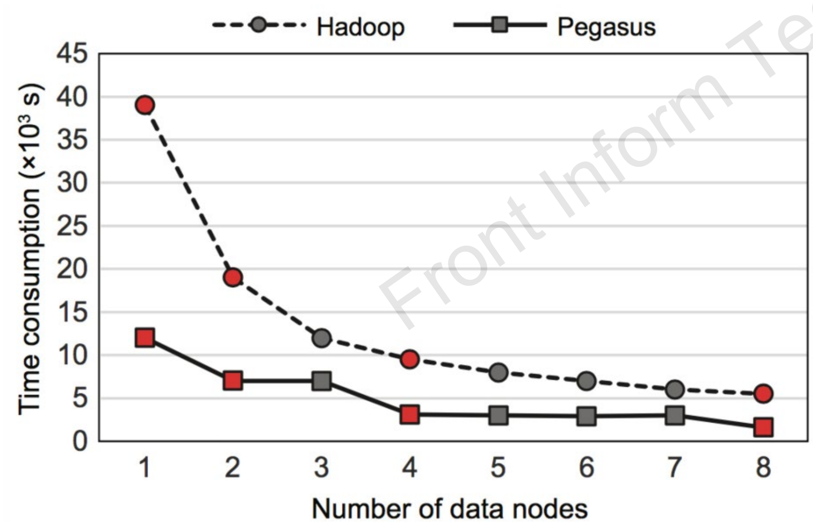

Fig. 6 Time consumption of Hadoop and Pegasus

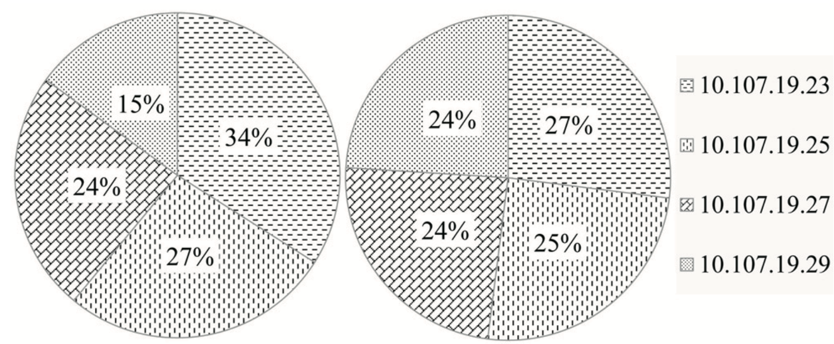

Fig. 7 Distributions of 1000 concurrent access on four mongos nodes: Hadoop (left) and Pegasus (right)

Major results (Cont'd)

- Our system is better load-balanced than default MongoDB.

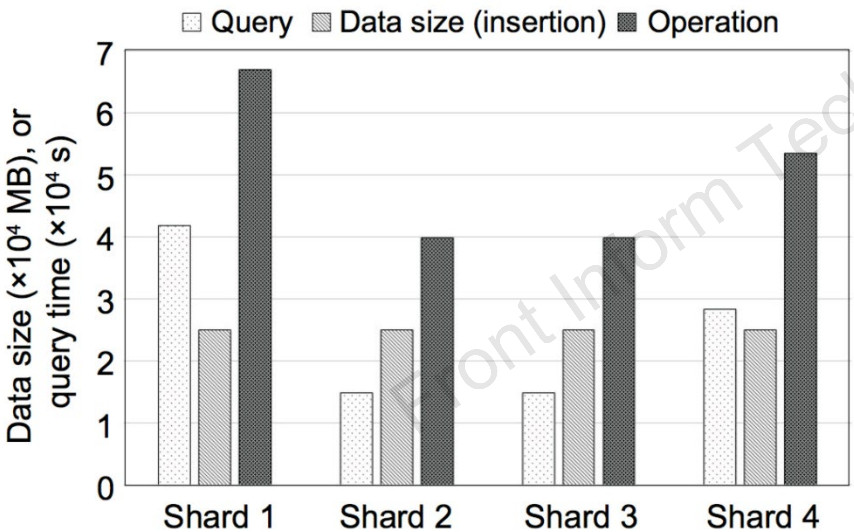

Fig. 8 Distributions of data size and query time with the default strategy in MongoDB

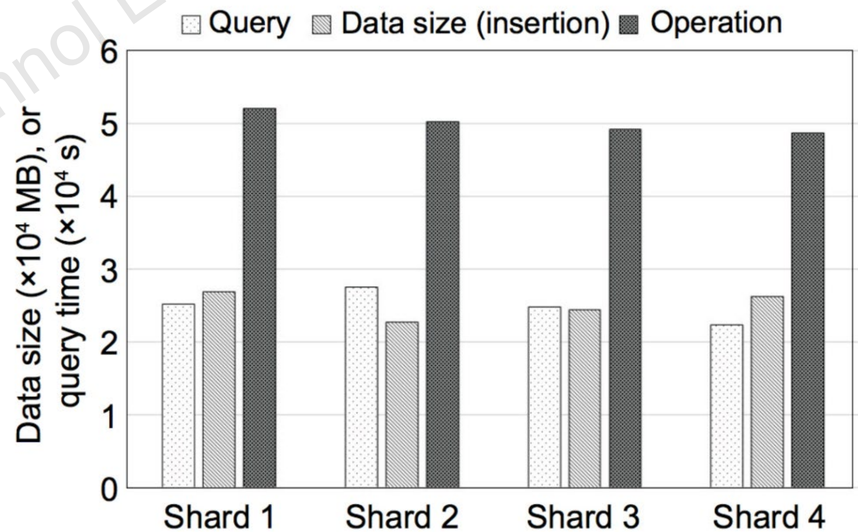

Fig. 9 Distributions of data size and query time with the optimized strategy in MongoDB

Major results (Con’d)

- Compared with DFIS, Pegasus realizes about 40% time reduction when transmitting data between its two subsystems.

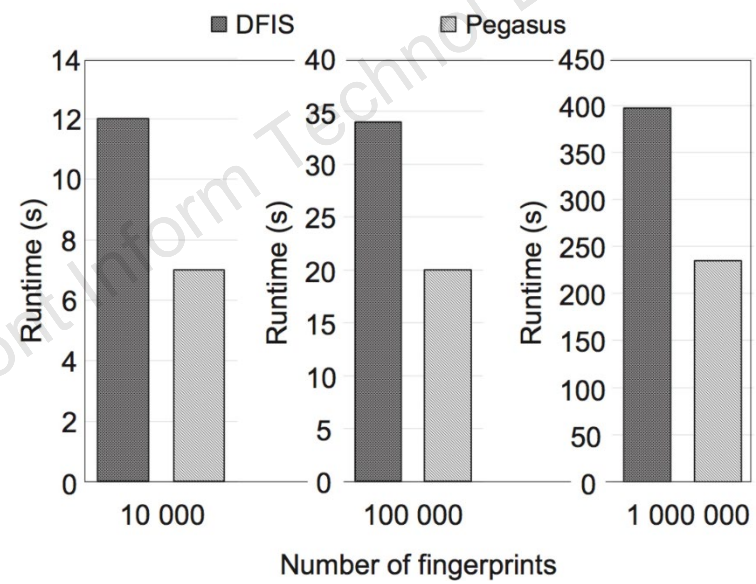

Fig. 10 Runtime comparison of data transmission between two subsystems on different datasets

Conclusions

- Compared with the Hadoop platform, Pegasus degrades the time consumption by almost 70%.
- Pegasus degrades the difference of access requests to less than 5%, which improves the stability of the distributed MongoDB tremendously.
- Without additional data operations (load and write) with HDFS, Pegasus reduces as much as 40% time consumption, compared with DFIS.