Redis + Twitter

Dennis Tsang
CPSC 416

Motivation

- Database-backed web applications require caching to increase performance

Solution?

- Memory is fast
- Cache results in memory

Strings

memcached

Integers

Pub/Sub

Strings

Transactions

Lists

redis

Sets

Sorted Sets

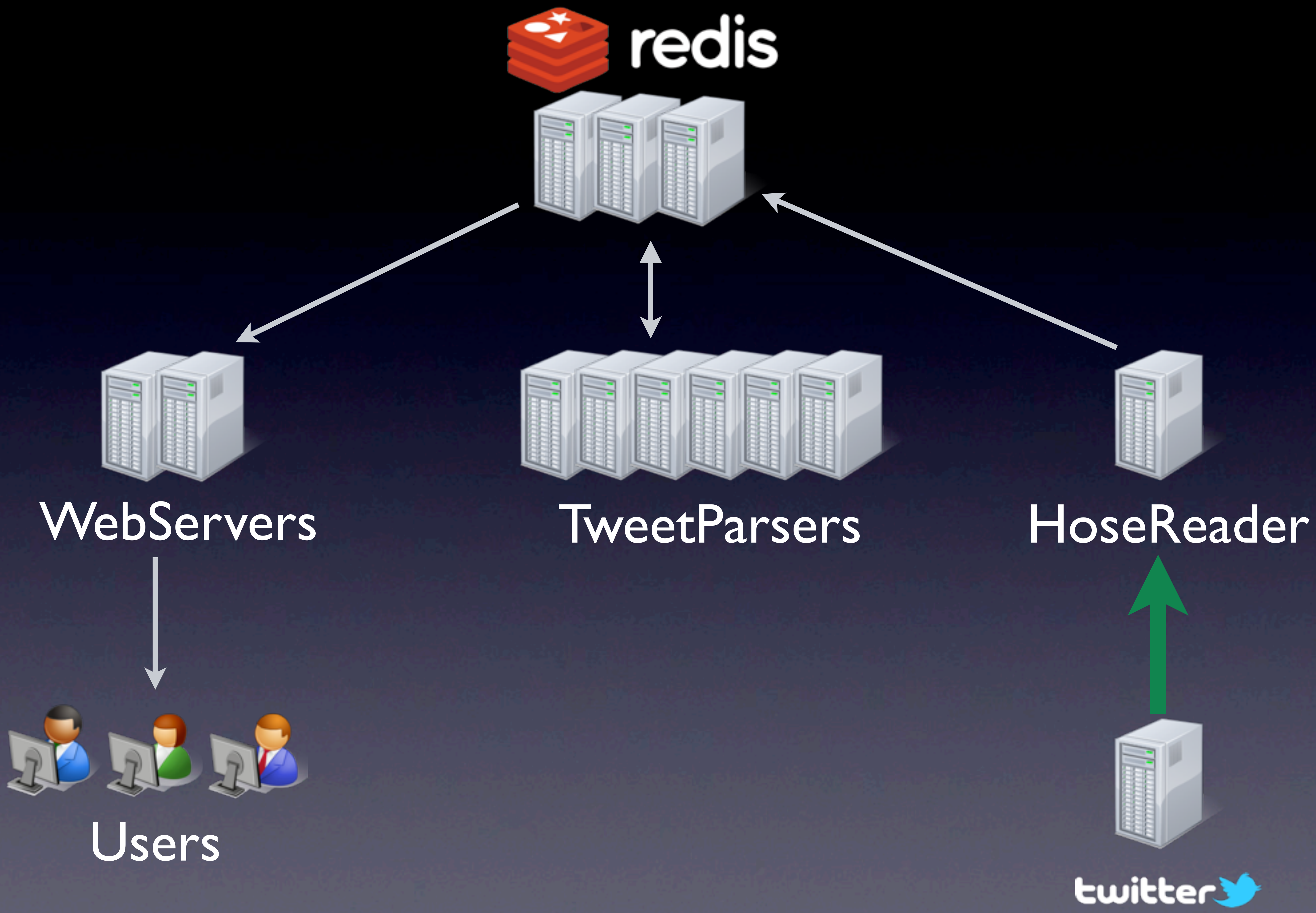
Integers

Hashes

- Beyond the key-value model
- Speed
- Persistent
- Master-slave replication

twitter.popular.words

- Grabs real-time tweets from Twitter’s “garden hose”
- Distributes processing over multiple Amazon EC2 instances
- Uses Redis to store all data
- Shows results through web interface



WebServers

TweetParsers

HoseReader

Users

Demo

To-Do & Possible Extensions

- Election of new master Redis server in the event the master fails
- Aggregation from other streams such as Google Buzz