



Cache Augmented SQL (CASQL) Systems

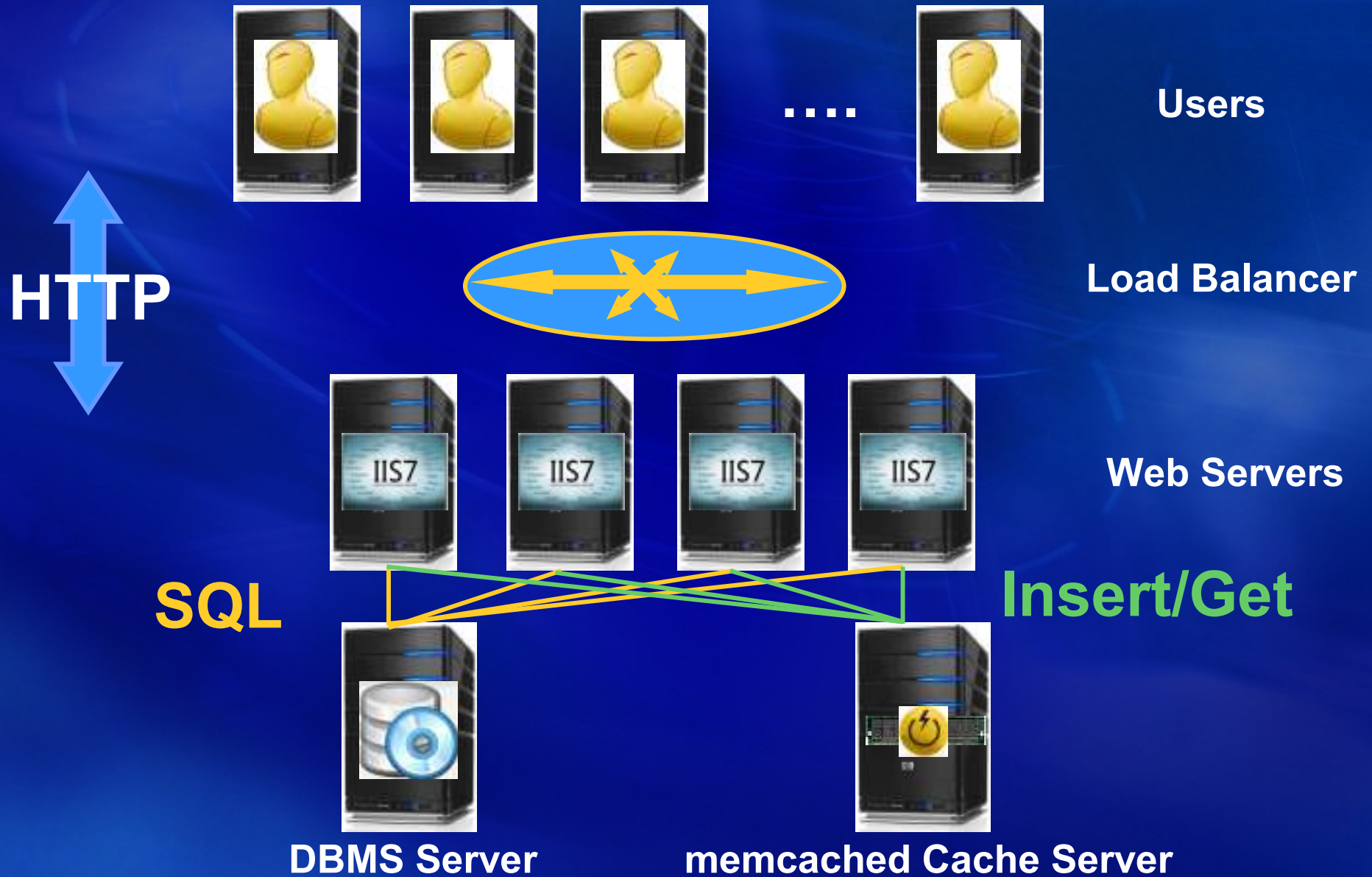
Shahram Ghandeharizadeh, Jason Yap
USC Database Lab

Interactive Social Actions

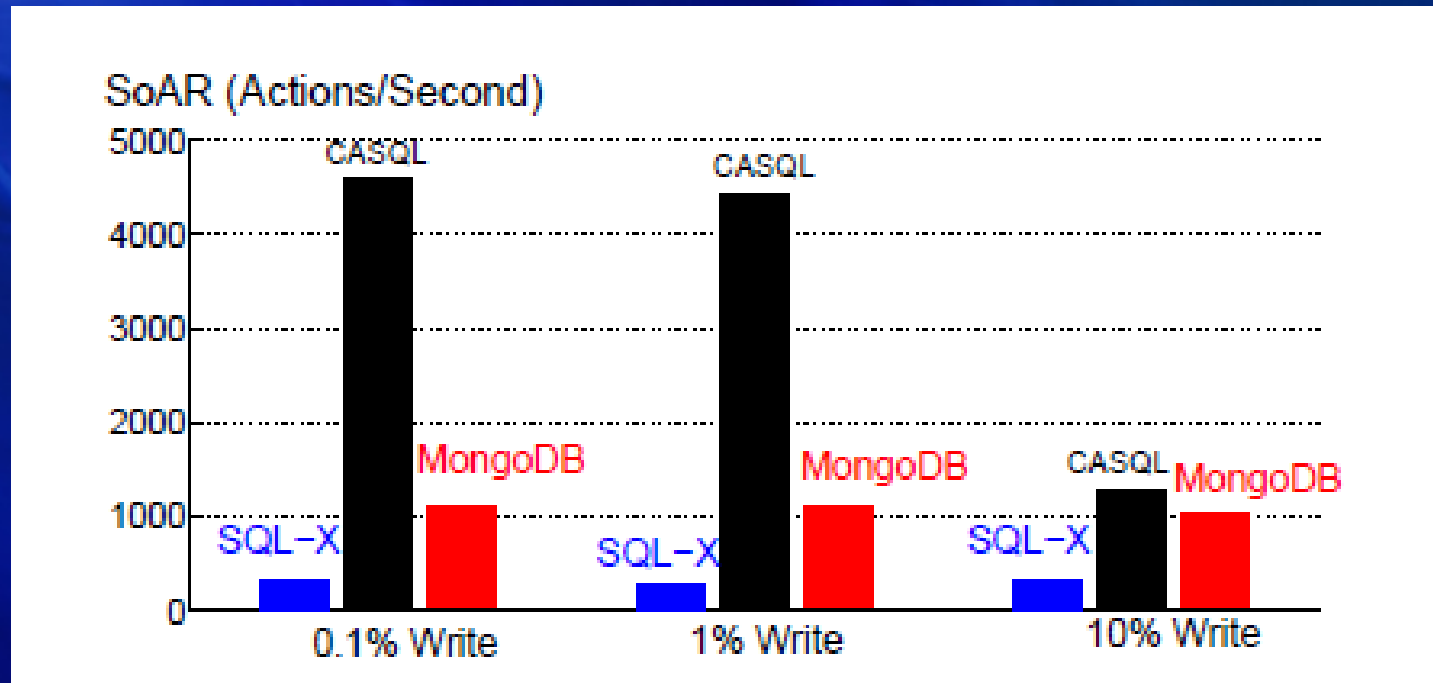
- **Read intensive workloads. Queries that compute the same result repeatedly.**
 - **Same result because the database changes infrequently, e.g., your Facebook account:**
 - **How often do you visit your Facebook profile page?**
 - **Once a second, every minute, every hour, once a week?**
 - **How often do you add and drop friends/change your profile?**



CASQL Architecture



Enhanced Performance



SLA: 95% of actions to observe a response time below 100 msecs

CASQL Today

Developer 1



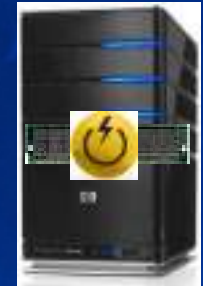
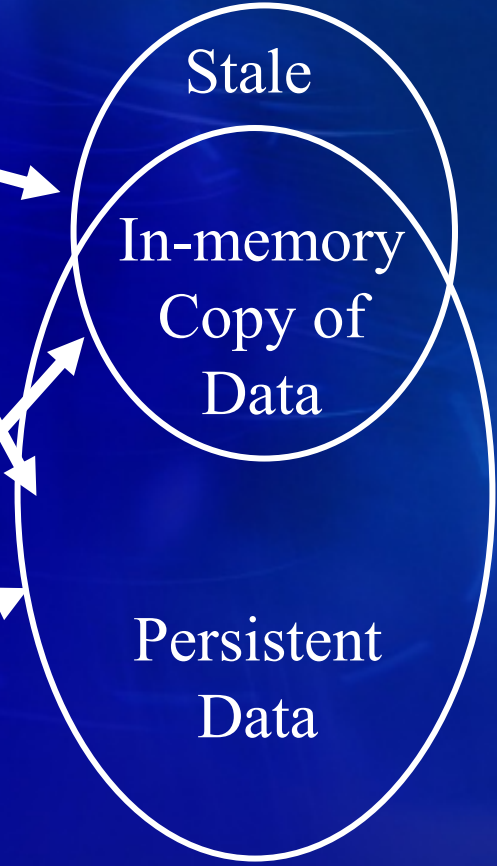
Application programs



Developer 2



Application programs



**memcached
Cache
Server**



**DBMS
Server**

Before DBMS: 1960/70s

Developer 1



Application programs



Developer 2

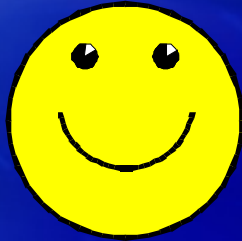


Application programs



CASQL DBMS

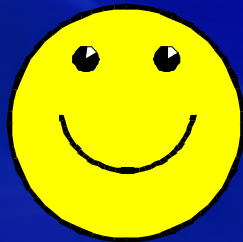
Developer 1



Application programs



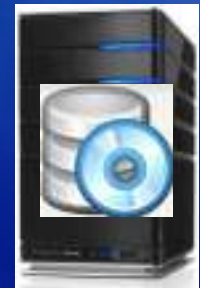
Key Value
Cache Server



Application programs



CASQL
DBMS



DBMS
Server

Physical Data Independence.

Stonebraker/Cattell's 10 Rules

1 Look for shared-nothing scalability.	2 High-level languages are GOOD and need not hurt performance.	3 Plan to carefully LEVERAGE main memory databases.
4 High availability and automatic recovery are essential for SO scalability.		5 ONLINE EVERYTHING.
6 Avoid MULTI-NODE operations.	7 Don't try to build ACID consistency yourself.	8 Look for administrative SIMPLICITY.
9 Pay attention to NODE performance.	10 OPEN SOURCE gives you more CONTROL over your future.	

Research Challenges

- **Physical data independence:**
 - **Application transparent approach to key-value consistency.**
- **Data availability in the presence of failures.**
- **Scalability.**
- **Elasticity.**

- References: S. Ghandeharizadeh and J. Yap. Cache Augmented Database Management Systems. USC Database Laboratory Tech Report 2012-07.