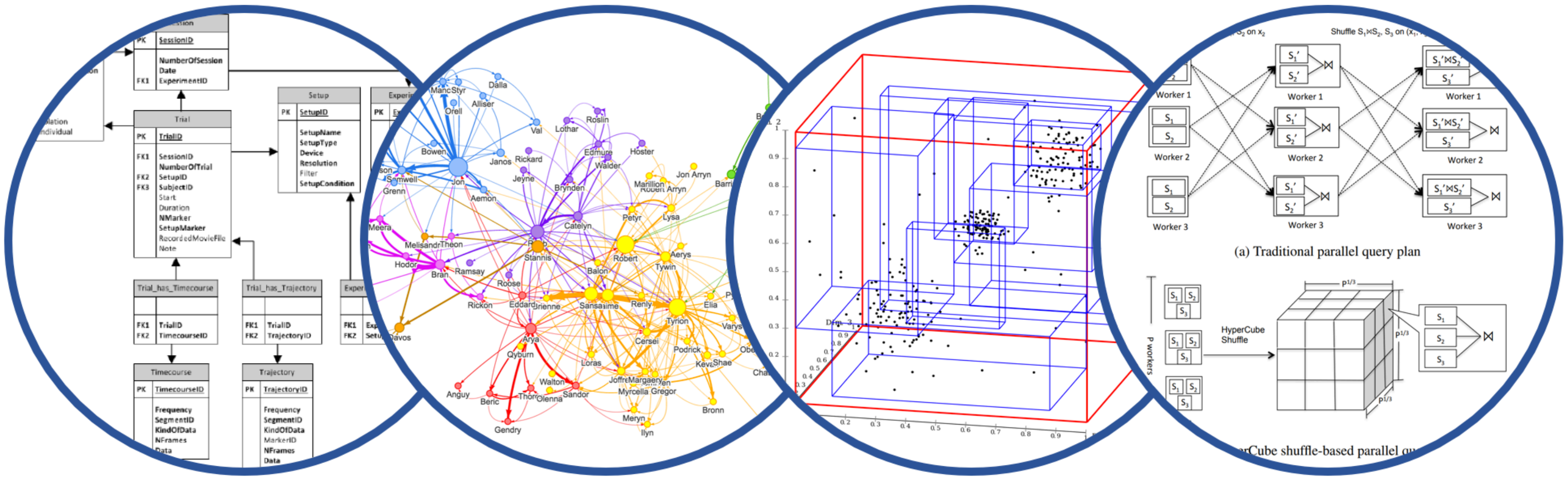


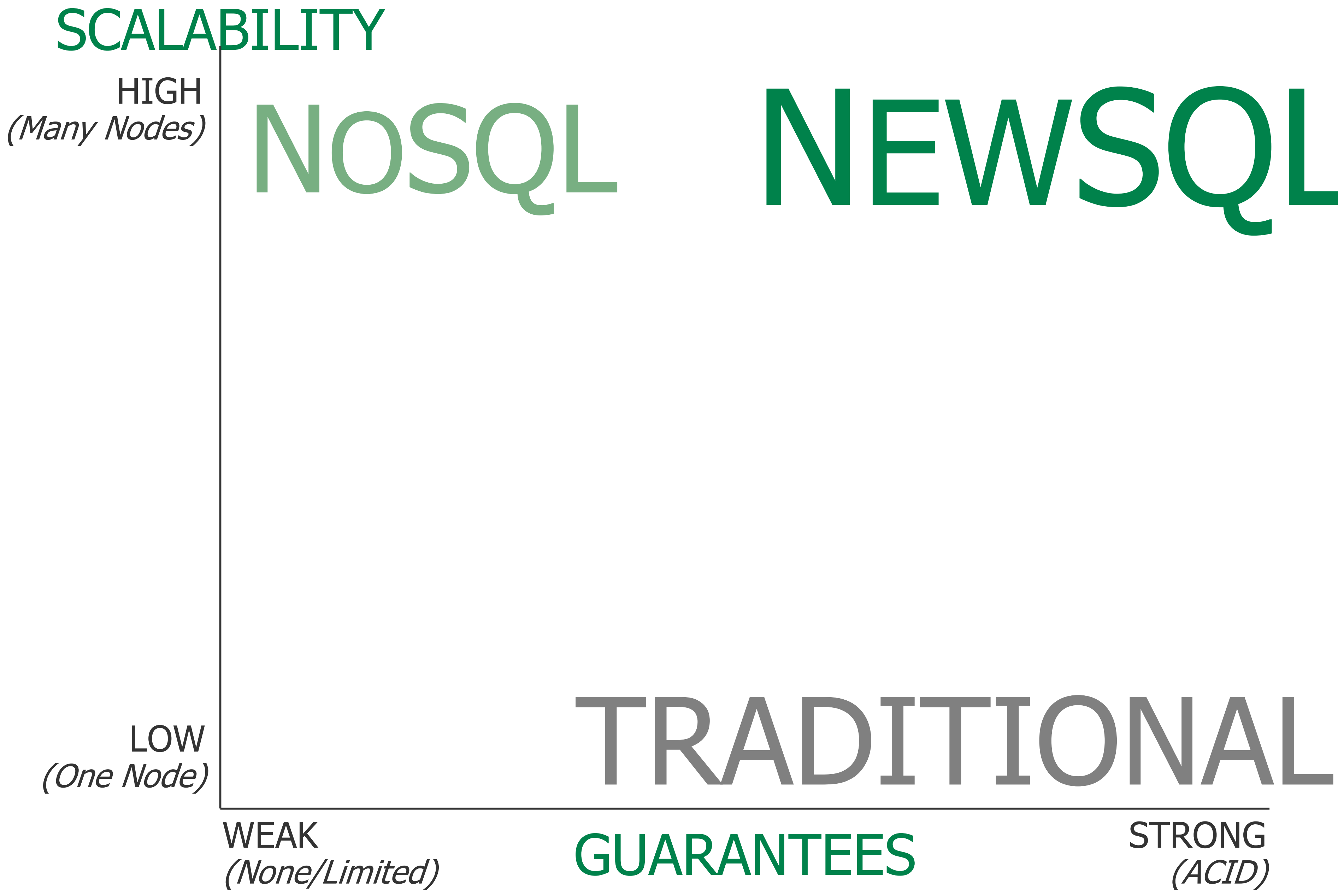
Database System Internals

NewSQL and Advanced Systems

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

Announcements

- Remember to do Quiz 2+3 on gradescope
- Course Evaluations are appreciated!
  - https://uw.iasystem.org/survey/242674



Slide from Andy Pavlo @ CMU

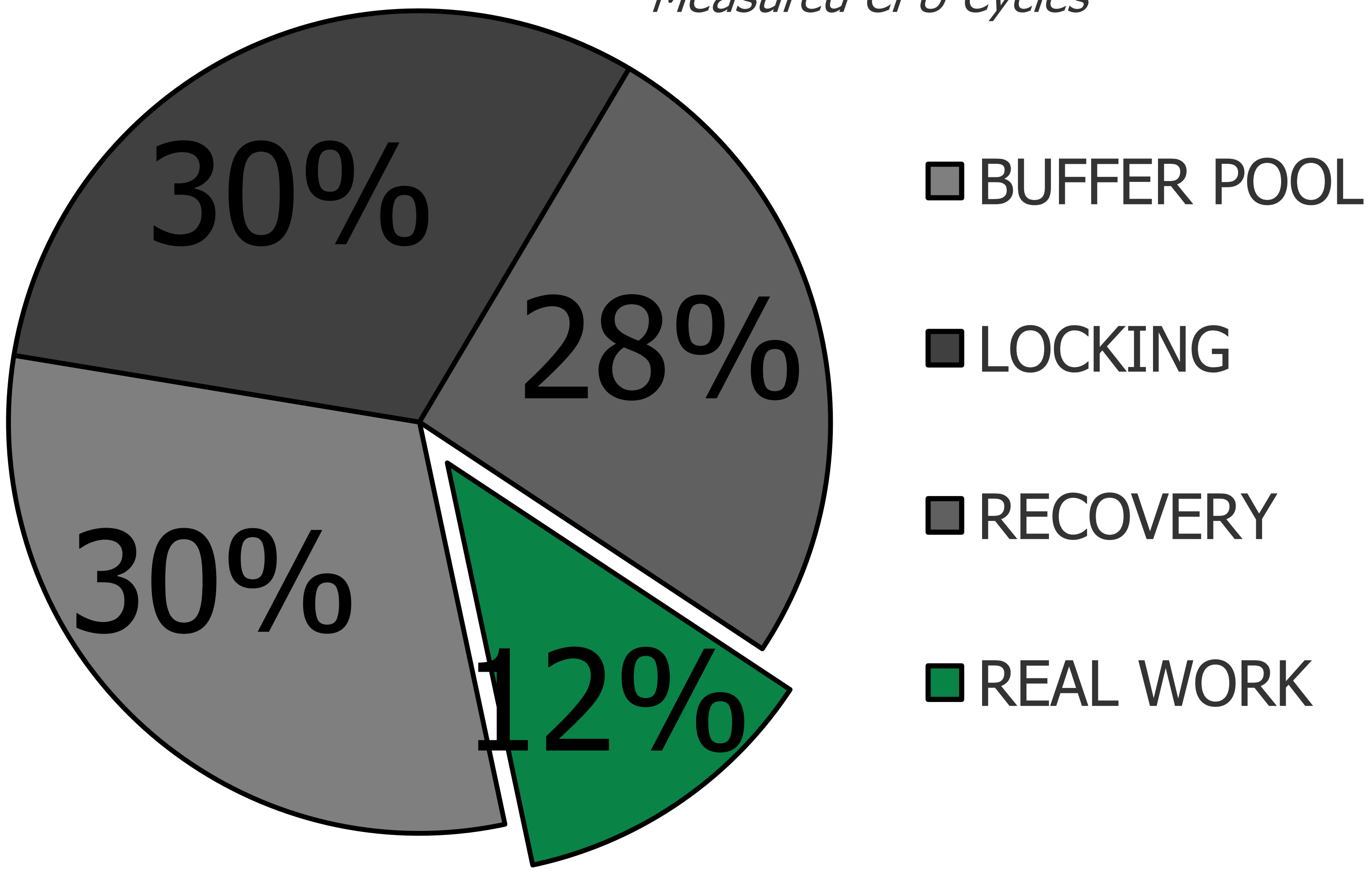
Some Popular NewSQL Systems

- H-Store
  - Research system from Brown U., MIT, CMU, and Yale
  - Commercialized as VoltDB
- Hekaton
  - Microsoft
  - Fully integrated into SQL Server
- Hyper
  - Hybrid OLTP/OLAP
  - Research system from TU Munich. Bought by Tableau
- Spanner
  - Google

H-Store Insight

TRADITIONAL DBMS:

Measured CPU Cycles



OLTP THROUGH THE LOOKING GLASS,
AND WHAT WE FOUND THERE
SIGMOD, pp. 981-992, 2008.

Slide from Andy Pavlo @ CMU

H-Store Key Ideas

- Main-memory storage
  - Avoids disk IO costs / buffer pool costs
  - Durability through snapshots + cmd log
  - Replication
- Serial execution
  - One database partition per thread on one core
  - Avoid overheads related to locking
- All transactions are stored procedures
  - Command logging avoids heavy recovery overheads
- Avoid distributed transactions
  - But when needed, run 2PC

STORED PROCEDURE

VoteCount:

```
SELECT COUNT(*)
  FROM votes
 WHERE phone_num = ?;
```

InsertVote:

```
INSERT INTO votes
 VALUES (?, ?, ?);
```

```
run(phoneNum, contestantId, currentTime) {
    result = execute(VoteCount, phoneNum);
    if (result > MAX_VOTES) {
        return (ERROR);
    }
    execute(InsertVote, phoneNum,
                        contestantId,
                        currentTime);
    return (SUCCESS);
}
```

Applica…

Slide from Andy Pavlo @ CMU

Some Details

At one node:

- Data is partitioned
- One database partition per thread on one core
- TXN receives a time stamp TS = serialization order
- TXN is assigned to a "base partition"; if data is need for other partitions, it sends requests there
- Partition managers order the requests based on TS. If conflict: abort, then restart (since stored procedure) with larger TS
- When a TXN has been granted locks at all partitions that it needs, then it can execute
- If more partitions are needed, then abort/restart

Some Details

Stored procedure

- TXN = One stored procedure
- Arbitrary Java code, BUT must be deterministic! No: call to the systems clock, random number generators, messages to other threads
- Have several parameterized queries, i.e. with '?'
- Several invocations of these queries are collected in a *batch*, then sent to the engine for execution
- If the batch requests data from a partition where the TXN does not have the lock: ABORT/RESTART
- Commit across multiple partitions: 2PC
- Command log: write just the procedure name plus parameters; only for committed TXN

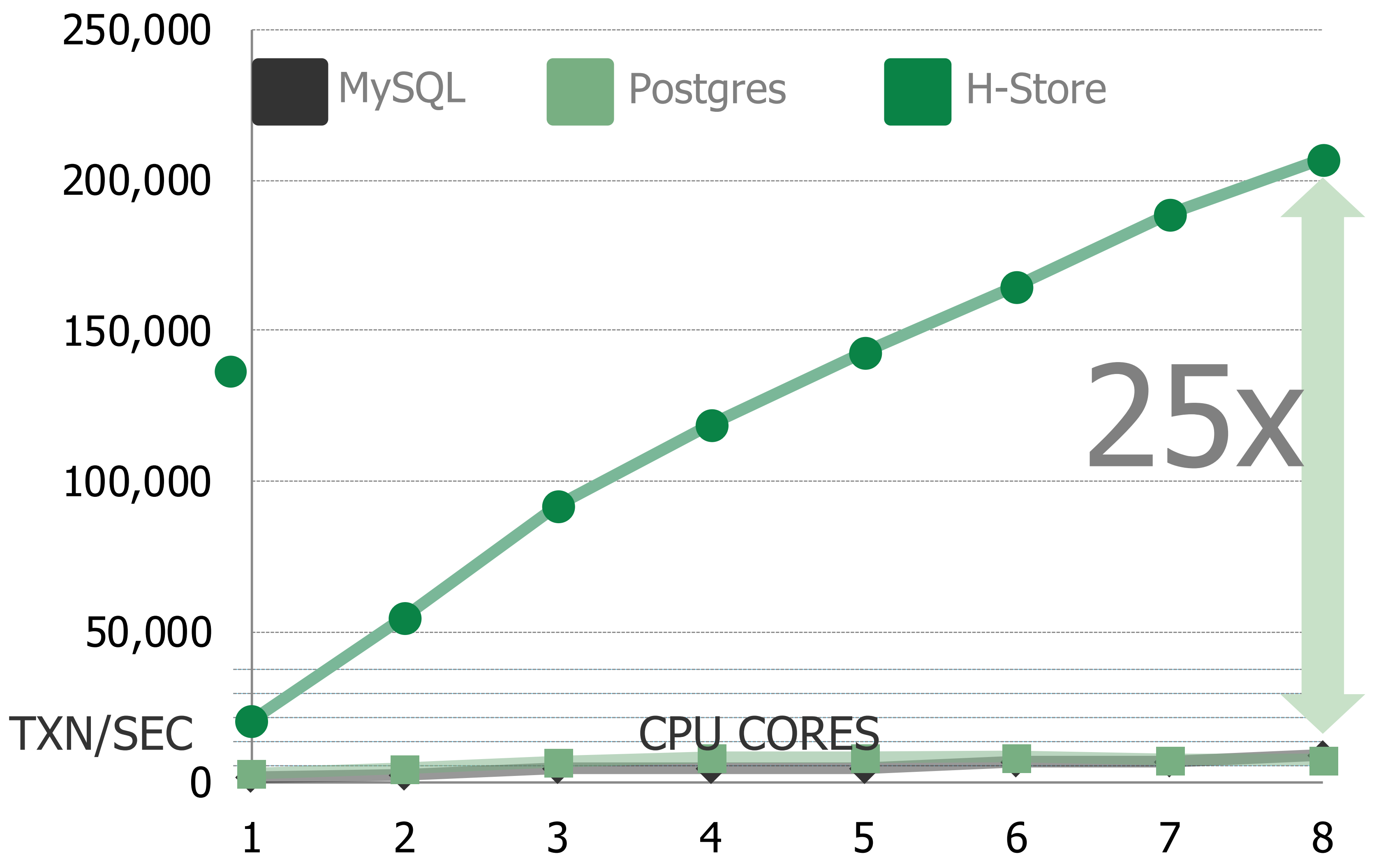
Some Details

Replication

- Recovery is slow → H-Store uses replication
- Initially, run Paxos to choose a master node
- During normal operation: TXN's are executed on the master node, who sends identical commands to the replica nodes; results are checked, and validated if majority, otherwise abort; minority nodes are considered failed
- When the master fails, run Paxos to elect new master.

Voter Benchmark

Japanese "American Idol"



Slide from Andy Pavlo @ CMU

Hekaton

- Focus: DBMS with large main memories and many core CPUs

- Integrated with SQL Server

- Key user-visible features
  - Simply declare a table "memory resident"
  - Hekaton tables are fully durable and transactional, though non-durable tables are also supported
  - Query can touch both Hekaton and regular tables

Hekaton Key Details

- Idea: To increase transaction throughput must decrease number of instructions / transaction
- Main-memory DBMS
  - Optimize indexes for memory-resident data
  - Durability by logging and checkpointing records to external storage
- No partitioning
  - Any thread can touch any row of any table
- No locking
  - Uses a new MVCC method for isolation

Hekaton More Details

- Optimized stored procedures
  - Compile statements and stored procedures into customized, highly efficient machine code

Hyper

- Hybrid OLTP and OLAP
- In-memory data management
  - Including optimized indexes for memory-resident data
  - Data compression for cold data
- Data-centric code generation
  - SQL translated to LLVM
- OLAP separated from OLTP using MVCC
- Exploits hardware transactional memory
- Data shuffling and distribution optimizations

Conclusion

- Many innovations recently in
  - Big data analytics
  - Transaction processing at very large scale
- Many more problems remain open
- This course teaches foundations
- Innovate with an open mind!