WiredTiger

Lightning Talk by:
Alex Degtiar
adegtiar@cmu.edu
What is WiredTiger?

NoSQL Storage Engine

- Underlying storage primitive of DBMS
- Local KVS
- e.g. Espresso and Tao

Different from many of the talks given here

- Data store is a building block, underlying component of all DBMS - Big Data
- Can be used separately as a local KVS
- Example: Espresso and TAO stores in MySQL(InnoDB) for Storage Layer
Why WiredTiger?

- Rethink engine for modern hardware
  - Many cores - scalability
  - Importance of RAM and I/O efficiency
  - Efficient power consumption

RAM - systems with more RAM, I/O expensive, try to maximize efficient use of memory
Underlying Storage

- **Row-oriented**
  - B-tree
  - write-optimized
- **Column-oriented**
  - store column groups in separate files
  - read-optimized
- **Log-structured merge trees, bloom filters**
  - write-optimized, sustained throughput

Mix and match, transaction
LSM same idea as BigTable (though LSM far predates it)
In-memory Optimizations

- Multi-core scalability
- Lock-free / non-blocking algorithms
- No in-place updates
- Maximize cache efficiency

Increasingly/efficiently utilize RAM compression, variable sized tree nodes..
On-disk Optimizations

● Efficient I/O
  ○ Compression
  ○ Column store
  ○ Large, variable-sized chunks
● LSM tree structures

Making I/O more valuable
Compression especially with column stores, which lend themselves to efficient I/O
LSM - SSD
Conclusions

- Modern NoSQL Storage Engine
- Utilizes many cores, RAM, and I/O efficiently
- Mixed storage formats (B-tree, Column, LSM)
- High performance vs. other engines

LevelDB, InnoDB
References

- WiredTiger website: [http://source.wiredtiger.com/1.4.2/architecture.html](http://source.wiredtiger.com/1.4.2/architecture.html)
- Benchmarks: [https://github.com/wiredtiger/wiredtiger/wiki/LevelDB-Benchmark](https://github.com/wiredtiger/wiredtiger/wiki/LevelDB-Benchmark)