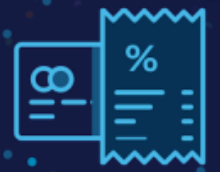
MEMSQL



A Trillion Rows Per Second as a Foundation for Interactive Analytics

Eric Hanson, Principal Product Manager

April 18, 2018

Overview

- MemSQL
- Interactivity and user satisfaction
- State-of-the-art query execution technology
- Demo
- Where can we go with this technology?

MemSQL Overview

What is MemSQL?

- SQL DBMS
- Fast: scale-out, compilation, in-memory, vectorized
- In-memory rowstore
- Disk-based columnstore
- Transactions and analytics
- Fantastic operational data store

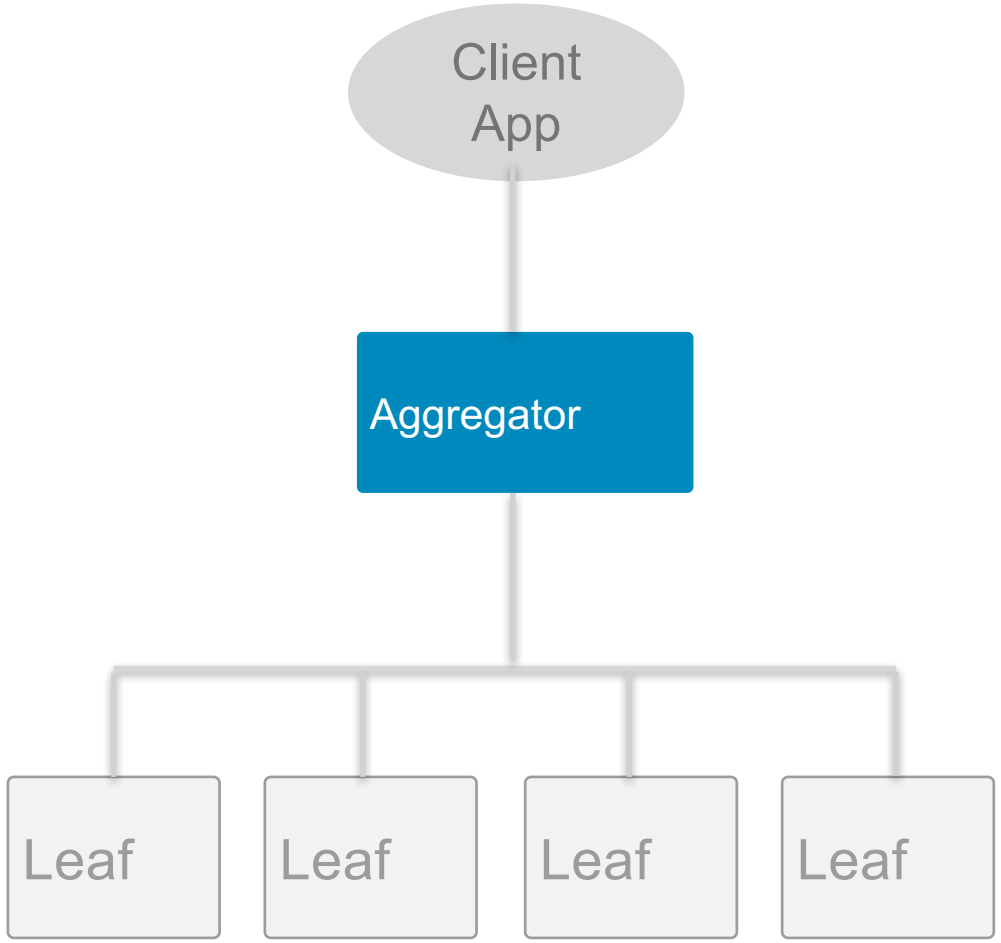
Why MemSQL?

FAST DATA
Ingest

LOW LATENCY
Queries

HIGH
Concurrency

MemSQL scale-out architecture



MEMSQL

Challenges to lightning-fast response

- Large data volume
- Many concurrent users
- Query complexity
- Rapidly changing data

Response Time, Productivity, and User Satisfaction

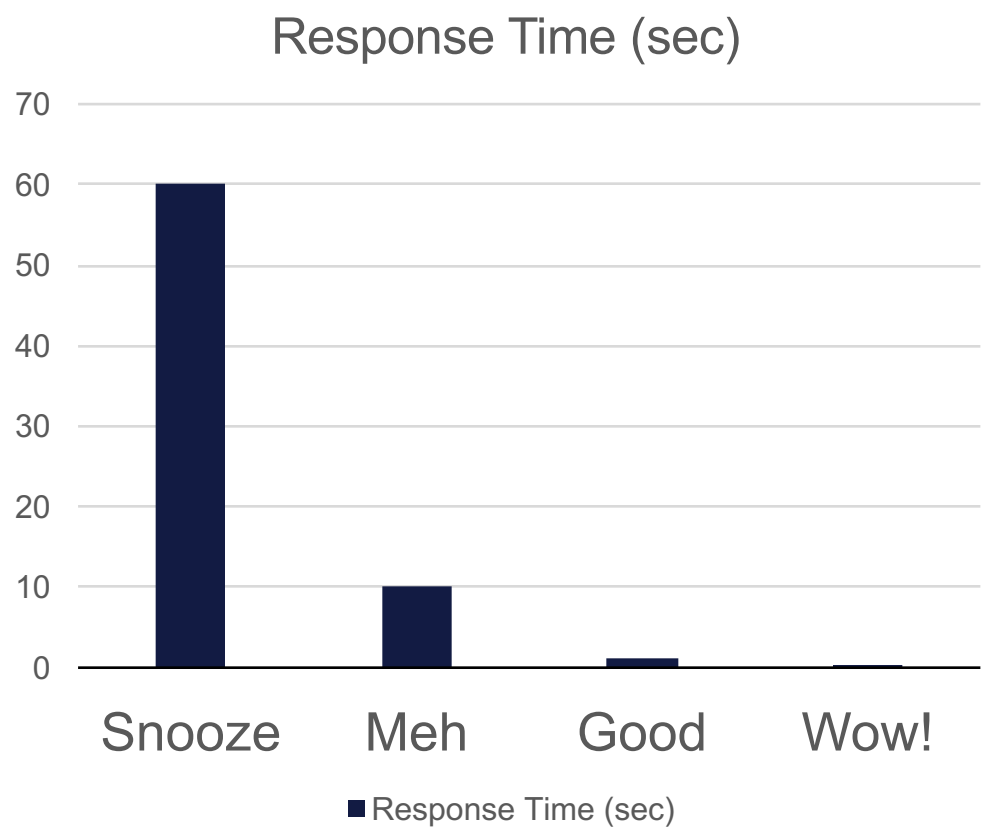
Stimulation is the indispensable requisite for pleasure in an experience, and the feeling of bare time is the least stimulating experience we can have.

WILLIAM JAMES,

1842-1910 Principles of Psychology, Volume I (1890)

The need for speed

- Users become used to fast response & expect it
- Satisfaction increases as response time decreases
- Delays over 50-150 msec are noticeable in realtime apps
- ~250 msec is median human reaction time



MEMSQL

Subtleties about response time

- High variance can bother users
  - < ¼ of mean or > 2X the mean
  - Can help to give message if high variance
- *Unexpectedly* fast results can make users apprehensive
- Fast response
  - Can lead to more “input errors”
  - *Makes users interact and explore more* ← Creates business value

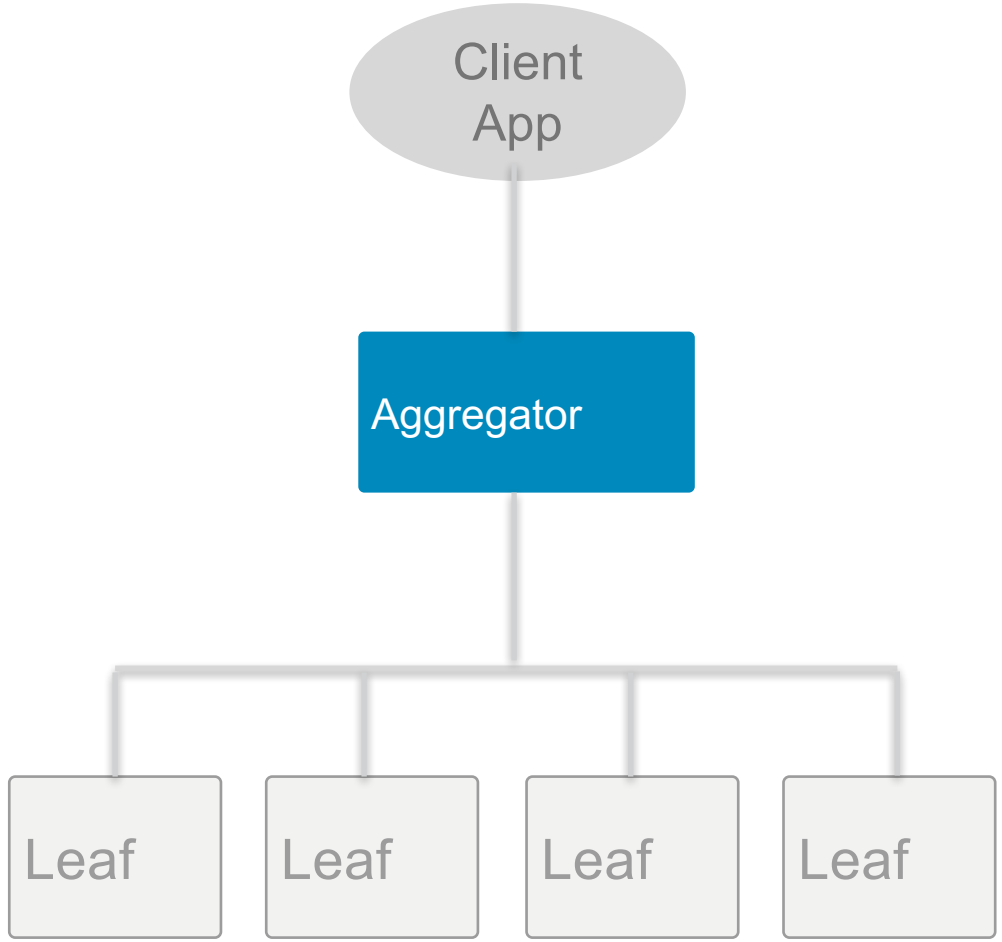
MemSQL Query Execution Technology

MemSQL technology to give lightning-fast response for analytics

- Scale-out
- Compiled query
- In-memory row store
- Columnstore
- Vectorization
- Intel AVX2 SIMD

MemSQL Scale-Out

- True horizontal scaling
  - Shared nothing
  - **Not** shared disk
- Hash partitioning across leaf nodes
- Can resize cluster and redistribute data
- Can add aggregators or leaves
- Scales both transactions and analytics



MEMSQL

MemSQL Compiles Queries

- Queries compile to machine code
- Example is **Row Store**
- First run takes compile time
- 49.3 million rows/sec on 2 cores
- 24.7 rows/sec/core
- Compare to 1 to 2 million rows/sec/core on interpreted DBMS

```
memsql> select count(*) from t;
+----------+
| count(*) |
+----------+
|  8388608 |
+----------+
1 row in set (0.10 sec)

memsql> select count(*) from t where color = "Red";
+----------+
| count(*) |
+----------+
|  4194304 |
+----------+
1 row in set (0.42 sec) ← includes compile time

memsql> select count(*) from t where color = "Red";
+----------+
| count(*) |
+----------+
|  4194304 |
+----------+
1 row in set (0.17 sec) ← executes from cache
```

MemSQL Columnstore

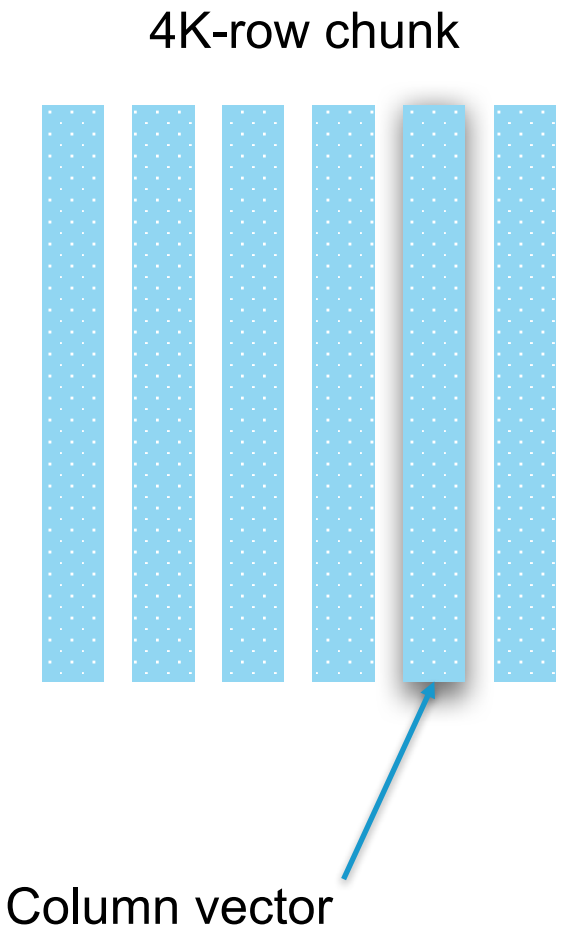
- On disk
- 1M-row segments
- Each column stored in separate file
- Only read columns you touch
- Highly compressed
  - Dictionary
  - Run-length
  - LZ
  - Integer value
- Min/max per column per segment

MemSQL columnstore ctd.

- Sorted by key
- Segment elimination
- Compiled code built into system for handling segments
- Linux file buffer caches keeps data in RAM
- In-memory row store segment for new data
- Background merger

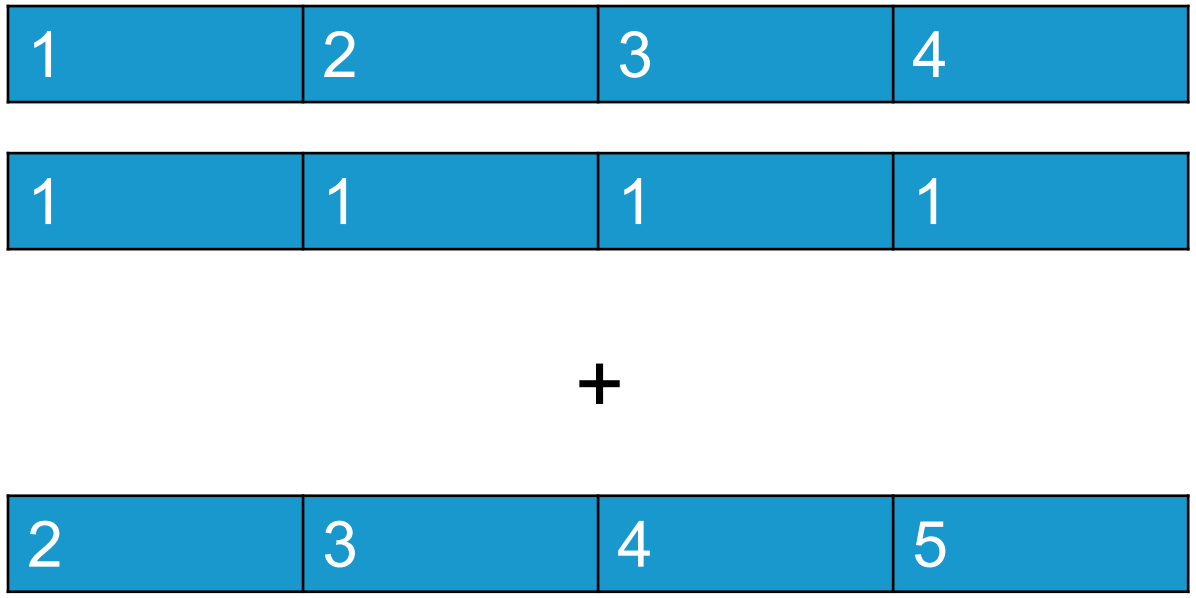
Vectorization

- Process data in 4,000-row chunks
- a.k.a. “vector projections”
- Process column vector in a tight loop of C++
  - Filters
  - Local group-by
  - Joins
- Few hundred million rows/sec/core



SIMD overview

- Intel AVX-2
- 256-bit registers
- Pack multiple values per register
- Special instructions for SIMD register operations
- Arithmetic, logic, load, store etc.
- Allows multiple operations in 1 instruction



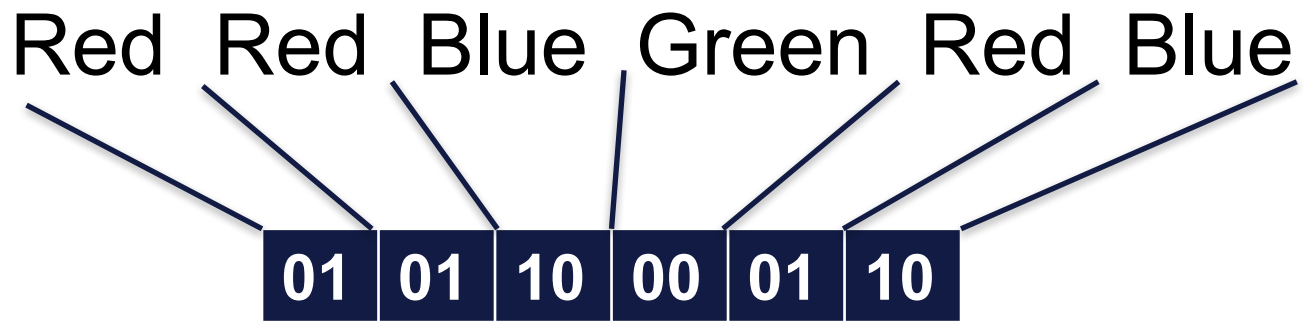
MEMSQL

Operations on Encoded Data in MemSQL

- Intel AVX-2 SIMD
- Filters
- Group-By
- Process 256-bit chunk of encoded (compressed) data at once
- Can process > 3 billion rows/sec/core
- Applied *before* vectorization for local group-by

Encoded data example

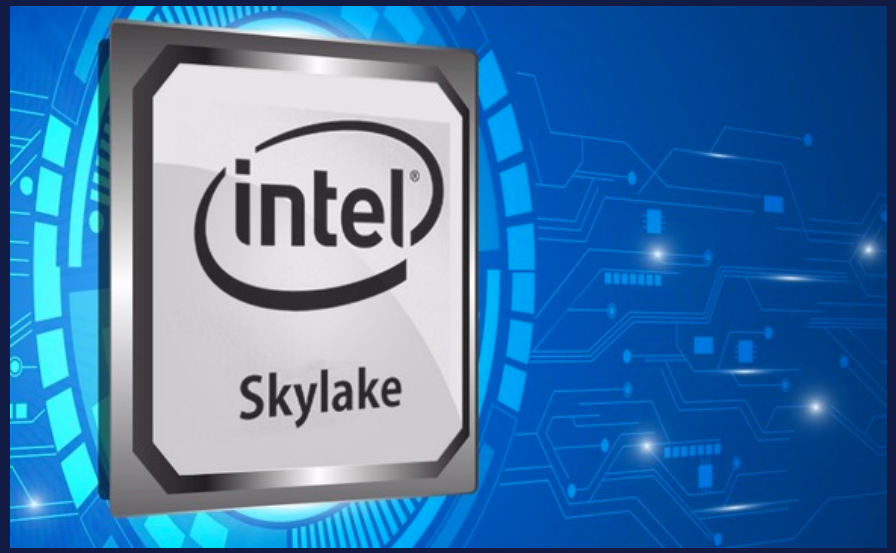
- Dictionary encoding
- Values:
  - Green: 00
  - Red: 01
  - Blue: 10
- ```
  select color, count(*)
  from t
  group by color
  ```



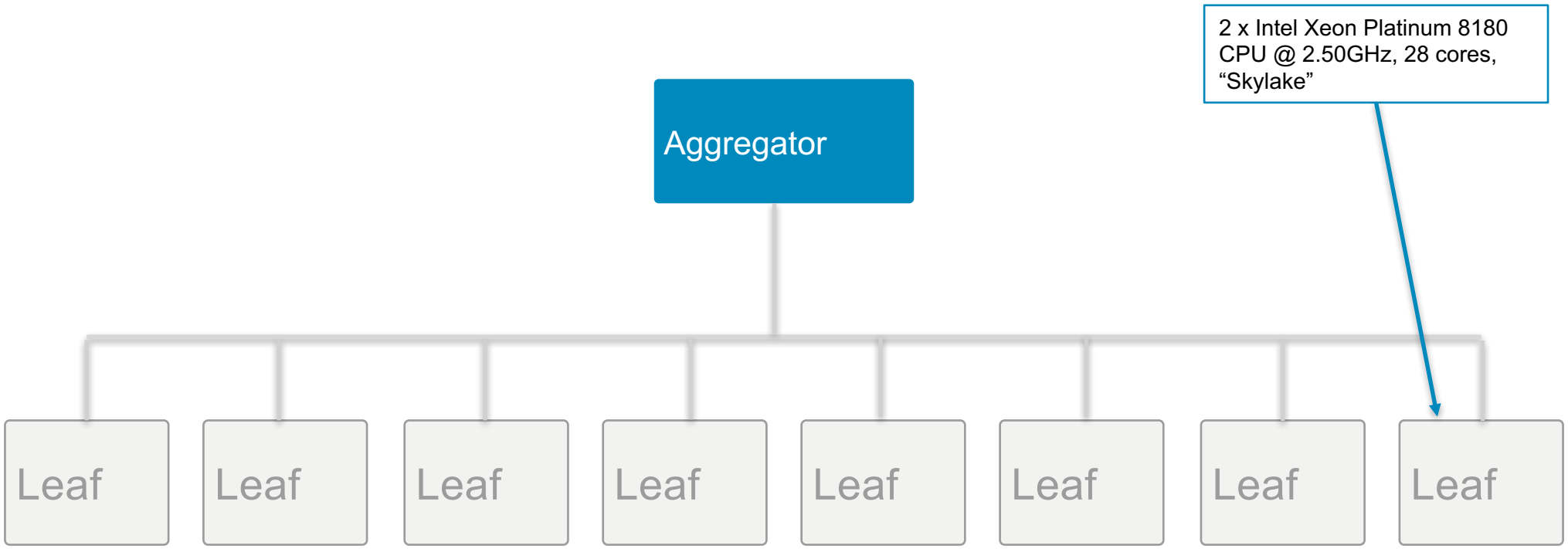
6 values in only 12 bits!

SIMD can process multiple 2-bit values at once
  ```

DEMO



The Hardware



Total leaf cores = 8 x 2 x 28 = 448

MEMSQL

The data

- Synthetically-generated stock trades
- 57.8 billion rows

MEMSQL

How big is a trillion?

Dollar amount of a football field covered with stacks of $100 bills 6 feet high

Number of tweets in 5 years

Number of text messages in the world in 45 days

More than the number of checkout transactions at Walmart since it was founded

Drum Roll Please!

The results

- Avg query time: 0.0525 sec
- 57.8 billion / 0.0525 =

1.10 trillion rows/sec

What does it mean?

- You can encourage analytic exploration
- The technology exists to meet these challenges:
  - Expectation of interactive response
  - Data explosion
  - Higher concurrency demands
  - Preference for SQL
  - Real-time update
  - Need to run on economical hardware

MEMSQL

Thank You!