Verizon Wireless - Fraud

Jan Shook, Principal
Oracle Open World 2012

Verizon Wireless Fraud

- Mitigate roaming (esp. international) usage fraud
- Problem: Proverbial needle-in-the-haystack
- Profile Call Detail Records (Billing)
- Billions of records / day
- Tens of thousands of records / sec
- Volumes drive IT Capital and Operational Expense

Legacy Footprint

- Oracle (Sun) SPARC III / IV
- 3 refrigerator-sized machines
- 6,000+ pounds, 100,000+ watts
- At 100% capacity (24x7)
- Oracle 10g, ~30 TB storage

← Pete (our Unix Admin)

Current Footprint

- Oracle (Sun) T5240
- 2 RU, 40 lbs, 1 kW
- At 20% capacity
- TimesTen, ~200 GB

← Pete's hand

The Pivot

Legacy / Vertical	Current / Horizontal
Store detail	Store summary
Storage drives design	Events drive design
Database drives performance	Latency drives performance
~30 TB storage	~200 GB storage
~100 billon of rows	~700 million rows
Oracle 11g, SAN, ETL	Oracle TimesTen, SSD, Real time

Focus on Throughput

- Pivot -> horizontal -> streaming -> throughput
- As throughput increases, latency must decrease
- Latency became a key design criterion
  - Application moved into a [soft] real time category
  - Problems: C++ operations failing at 180 to 200 us
  - Average op window < 100 us

Technology Stack

- Oracle (Sun) T-series processors
  - T1 (T2000), T2+ (T5240), … now T4 (T4-2)
  - 32 GB memory, then 256 GB, … now 512 GB
  - Direct attached storage, SAN, … now all flash
- Custom multithreaded, pipeline application
  - C/C++, ~400k LOC, ~52 stages (thread pools)
  - Memory maps (in-memory “database”)

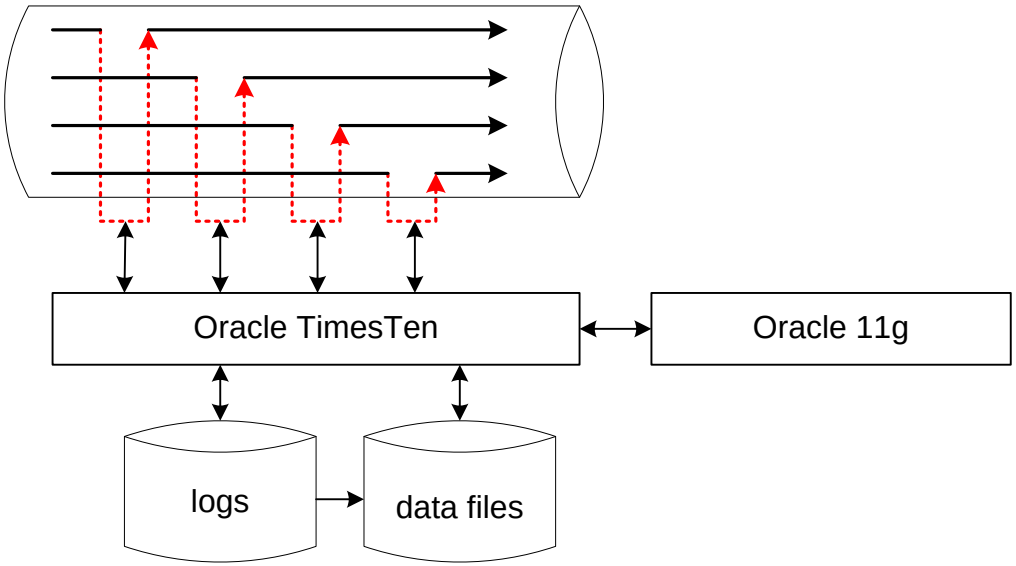
Home-grown In-memory Solution

- Started with large memory maps (>50 GB)
- Success with fixed-size mmaps (Solaris is very efficient)
- Learned that growing the data set is very hard
  - Tried partitioning: Fixed + Variable
    - Large fixed partition (mmap) with binary search (no index)
    - Small variable partition with indexes, random I/O, SSDs, …
  - Worked for lower volumes, but failed at higher volumes

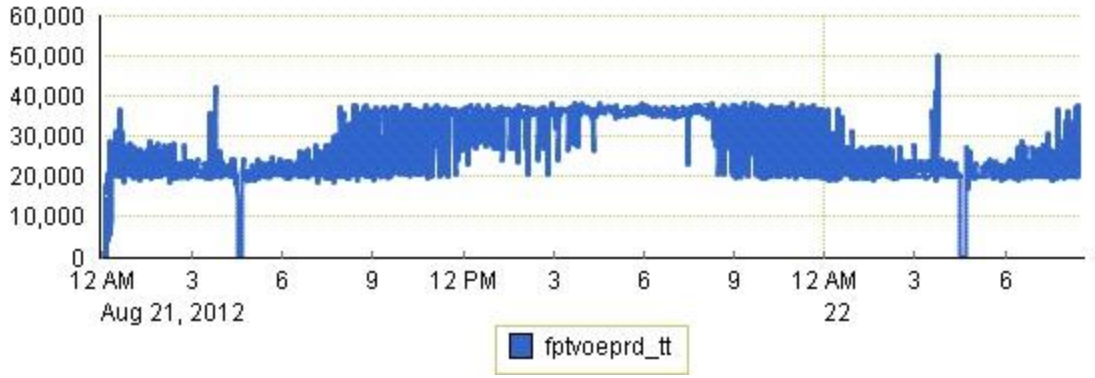
TimesTen In-memory Database

- Significant benefits
  - Persistence: Moves in-memory data to disk
  - Ops are within required tolerance (<100 us)
  - IMDB Cache: Replicating Oracle to TimesTen (vice versa)
  - OEM: Instrumentation
- Challenges
  - ODBC native interface
  - Transactional vs. throughput-oriented threading model

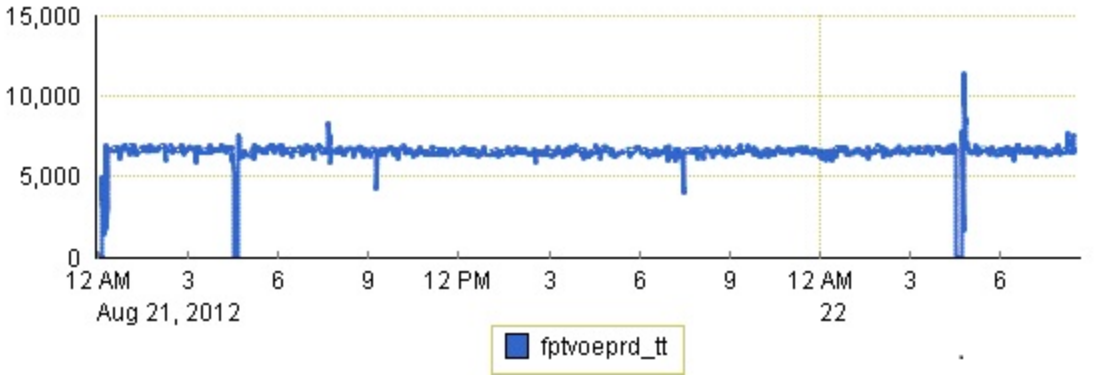
Throughput vs. Transactions



TimesTen Selects & Merges



~40k Selects / sec



~7k Merges / sec

Final Thoughts

- TimesTen does what it's supposed to do
  - It's a database: Application still responsible for design
- Design is critical: Threading, latency, replication
- We're looking forward to HA and DR projects

