HYBRID TRANSACTION/ANALYTICAL PROCESSING

COLIN MACNAUGHTON
WHO IS NEEVE RESEARCH?

- Headquartered in Silicon Valley
- Creators of the X Platform™- Memory Oriented Application Platform
- Passionate about high performance computing
- Running in production at Fortune 100-300
What is HTAP … What are the Challenges?
How The X Platform tackles HTAP
HTAP Use cases
WHAT IS HTAP?

Hybrid transaction/analytical processing will empower application leaders to innovate via greater situation awareness and improved business agility. This will entail an upheaval in the established architectures, technologies and skills driven by use of in-memory computing technologies as enablers.

- Gartner 2014

HTAP allows businesses to react to “business moments” … transient opportunities and risks that exist in the now.
TYPES OF APPLICATIONS

- Credit Card Processors
- Personalization Engines
- Ad Exchanges
- IoT Event Processors
- Financial Trading Risk Engines
- …
WHAT DO WE NEED?

- **Performance**
  - 100s of thousands of transactions a second
  - Microseconds to low milliseconds processing times

- **Scale**
  - 10s of millions of records in application’s working set
  - Scale linearly *with the business*

- **Reliability / Availability**
  - Zero message or data loss across network, process, machine or data center failures

- **Agility / Ease**
  - Write pure Java business logic without concern for above, ability to evolve applications organically

- **Intelligence**
  - Ability to analyze working state and absorb streaming intelligence *quickly* to react to business opportunity and risk.
A SIMPLE ARCHITECTURE (UNTENABLE)

Transaction Processing Apps (OLTP)

- Application

Enterprise Data

- Relational DB

Update Intensive, Short Transactions

Read Intensive, Long Transactions

Analytical Processing (OLAP)

- Analytics

Requirements:
- Scale
- Performance
- Reliability
- Agility
- Intelligence

Request Stream

Choke Point:
Long running OLAP queries Starve OLTP Business Transactions

Requirements:
- Visualization
- Capture
- Aggregation/Transformation
- *Timely* BI Feedback
THE TRADITIONAL ARCHITECTURE (ETL)

- **Transaction Processing (OLTP)**
  - Application
  - Operational Database
  - Request Stream

- **Data Integration (Extract, Transform, Load)**
  - Data Warehouse

- **Analytical Processing (OLAP)**
  - Analytics

**Benefits of ETL**
- Faster: Analyticals Decoupled
- Difficult to Scale (Update Contention)
- Complex

**Challenges of ETL**
- ETL allows OLAP without Compromising OLTP
- DataDuplication
- Slow (batch processing)

**Slow**
Analytical Feedback in Hours or even Days -> “Business Moment” Missed
ETL FAILINGS

- **Scalability**
  - Update Contention in Operational Database impedes scale

- **Performance**
  - Database read/write round trip latency impedes ability to stream.
  - Extract/Transform/Load is slow to avoid impacting operational data -> “business moment” is long gone by time analytics yield results.

- **Agility**
  - Data duplication due to mismatch between operational state and data warehouse.
  - ETL process is complex leading to fear about changing data warehouse schema and hampers innovation in transactional business logic.
HTAP DATABASES

Use In-Memory Technologies and Multi-Version Concurrency Control to allow transaction processing and analytical Loads on the same database.
ENTER HTAP DATABASES

Transaction Processing (OLTP)

Enterprise State

Leverages In Memory State
(faster updates/read)
+ MVCC -> concurrent OLTP/OLAP

Analytical Processing (OLAP)

✓ Eliminate Data Duplication
✓ Reduced Complexity
✓ Adoption Challenges?
  - who owns the schema?

Scaling Challenges: better, but still update contention

Mapping of objects to shared schema impedes agility

Request Stream

Application

HTAP DB

VoltDB, NuoDB, MemSQL…

Analytics

Analytics

Much more timely analytical Feedback
MICROSERVICES

Decompose Applications Into Individual Services that Perform Business Functions around State Private to that Service. With Inter-Service Collaborate Purely Over Messaging. Applications Can Then Scale By Partitioning of State.
SCALING OUT – STRIPED DATA + SMART ROUTING

Transaction Processing (OLTP)

Service A Partition 1

Service A Partition 2

Data “Striped”

Request Traffic

Enterprise State

HTAP DB

HTAP DB

Smart Routing

(messaging traffic partitioned to align with data partitions)

Analytical Processing (OLAP)

Analytics

Analytics

Analytics Results Streamed Back to Transaction Processors
Scalability
✓ Update contention handled by microservices and data striping.
-- Still some complexity in scaling data tier and transaction processing tier

Performance
✓ Ability to perform analytics without impacting OLTP

Transaction Processing Performance not optimal due to remote state. Have to scale very wide to absorb analytics streams

Agility
✓ Microservices allows more agile, lower risk delivery
-- Unclear who owns database schema when database is doing double duty for analytics and transaction processing.
-- Complexity mapping application state to database schema.
The X Platform is a memory oriented platform for building *multi-agent, transactional* applications.

Collocated State + Business Logic = Full Promise of In-Memory Computing
 ✓ Message Driven
 ✓ Stateful - 100% In Memory
 ✓ Multi-Agent
 ✓ Totally Available
 ✓ Horizontally Scalable
 ✓ Ultra Performant
EXTREMELY SIMPLE PROGRAMMING MODEL

MESSAGES

```xml
<messages>
...<messages>
<message name="MyInboundMessage">...
<field name="value" type="Long"/>
</message>
</messages>
</entities>
</model>
```

STATE

```xml
<model>
...<entities>
<entity name="MyAppState">
<field name="counter" type="Long"/>
</entity>
</entities>
</model>
```

MESSAGE HANDLERS

```java
@EventHandler
public void onMessage(MyInboundMessage message, MyAppState state) {
long counter = state.getCounter();
counter += message.getValue();
state.setCounter(counter);
MyOutboundMessage out = MyOutboundMessage.create();
this.messageSender.send(out);
}
```

✓ Scales horizontally
✓ Incredibly Fast
✓ Fault tolerant
✓ Zero Garbage

BUILD-TIME CODE GENERATION

✓ Built-In Schema Evolution
✓ Single Thread Handler Logic
✓ Provider Agnostic Messaging
✓ Transparent State Replication
✓ Exactly Once Atomic Handling

BUILT-TIME CODE GENERATION
HTAP WITH X – IN TRANSACTION ANALYTICS

Transaction Processing + In Transaction Analytics

Data “Striped”

Request Traffic

100% In Memory State As Java Objects

Service A Partition 1

Journal Storage 1 2 ...

Service A Partition 2

Journal Storage 1 2 ...

Smart Routing (messaging traffic partitioned to align with data partitions)

Async, Transactionally Consistent Change Data Capture

Journal Storage

Analytics

Analytics

HTAP DB

Analytics Results Streamed Back to Transaction Processors
**X PLATFORM - RELIABILITY**

**Primarily**
- Fast
- Durable
- Consistent
- Scales
- Simple

- In Application Memory Replicated + Partitioned
  - Application State fully in Local Memory
  - Pipelined Stabilization
  - Pure domain code
  - Single-Threaded Dispatch
  - Processing Swim-lanes
  - Smart Routing (messaging traffic partitioned to align with data partitions)

- Operate at memory speeds
- Plumbing free domain
- Scales with size and volume
X PLATFORM FOR HTAP- REPORT CARD

Scalability
✓ Update contention handled by microservices and data striping
✓ Single scaling metric: state scales with application

Performance
✓ Maximum throughput since state is local to function
✓ Local state allows in transaction analytics
✓ Change Data Capture allows asynchronous, optionally conflated

Reliability / Availability
✓ Pipelined Replication to Hot Backup(s),
✓ Journaled Storage, Change Data Capture to

Agility
✓ Microservices allows more agile, lower risk delivery
✓ Fire and Forget Messaging, Objects Transparently Persisted, Atomic
✓ Pure Business Logic, no infrastructure bleed
**REAL LIFE USE CASES**

- **MGM Resorts International**
  - eCommerce Engine is authored on the X Platform
    - 10 services/26 agents comprise the eCommerce service suite
  - Key metrics
    - All state, reference and transactional fully in-memory: ~1TB of in-memory state
    - Low 10s of millisecond catalogue/pricing update latency
    - Full 14 month dynamic pricing response time to website
    - Sub-second rate update to partner (wan)
  - SSO storage engine authored on the X Platform
    - Authored as a distributed, persistent, partitioned hash map
      - **Authored on X in 3 hours!**
    - <10ms response times @ 20k updates per second
      - Bottleneck in messaging bus, X has plenty of more capacity
200k Merchants
40k Card Holders
80k Cards
1 Year Card History
Only 2 partitions per agent
All agents running on just 2 servers
7,500 auth/sec, Full HA + X-Once

Auth Response Time = 1.2ms
Getting Started Guide
https://docs.neevereresearch.com

Reference Applications
https://github.com/neevereresearch/nvx-apps

We’re Listening
contact@neevereresearch.com
Questions ?