TimeBase
Introduction to TimeBase Community Edition

Presenters:

Ilya Gorelik –
VP, Real-Time Computing Lab, EPAM
Ilya_Gorelik@epam.com

Aliaksei Vavilau –
Chief Architect, TimeBase,
EPAM
Aliaksei_Vavilau@epam.com

https://timebase.info/
Agenda

Presentation:
• Overview & History
• Architecture
• Technical details

Demonstrations:
• TimeBase Web Admin
• Examples Use-Cases
  • Portfolio Risk Monitor
  • Instant Payment Dashboard
• QuantHub Custom Research Environment
• Grafana and Jupyter (Source code round-trip)

Q&A
Overview

• **TimeBase** is a high-performance *time series* database and *streaming system* developed by EPAM Real Time Computing Lab (formerly Deltix, Inc.).

• Current TimeBase is a result of **15 years** of experience in financial domain. TimeBase runs standalone or in a cluster, processes millions of messages per second, stores terabytes of data, and can offer low microsecond latencies.

History

• 2006 – TimeBase 1.0 (historical data analytics, SQL)
• 2010 – TimeBase 2.0 (timeseries data base, live streaming)
• 2015 – TimeBase 5.0 (very large databases, HDFS)
• 2018 – IPC/UDP low latency mode (TimeBase topics)
• 2019 – Cloud support (REST/WS, Docker, Kubernetes)
• **2020** – TimeBase “Community Edition” (Open Source version); public website [Timebase.info](http://Timebase.info) launched
## RTC Lab solution for advanced systematic trading

<table>
<thead>
<tr>
<th>TimeBase</th>
<th>QuantOffice</th>
<th>QuantServer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATA ACQUISITION &amp; PREPARATION</strong></td>
<td><strong>ALPHA DISCOVERY</strong></td>
<td><strong>SIMULATION WITH LIVE DATA</strong></td>
</tr>
<tr>
<td>Acquiring market and other types of data for model creation and testing</td>
<td>Creation of initial alpha model, based on analysis and manipulation of data</td>
<td>Deploying quant research model into production live trading environment</td>
</tr>
<tr>
<td></td>
<td>Back-testing of model using historical market data and other data sets</td>
<td><strong>ANALYSIS &amp; OPTIMIZATION</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Analyzing the results of back-testing and optimizing for improved results</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PRODUCTION</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simulation with live data and analyzing effectiveness of strategy</td>
</tr>
</tbody>
</table>

Source: Aite Group
## RTC Lab at a glance: < aggregate | analyze | act >

### EPAM TECHNOLOGY

<table>
<thead>
<tr>
<th>CORE TASKS AND ACTIVITIES</th>
<th>TimeBase</th>
<th>QuantOffice</th>
<th>QuantServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect and aggregate data from many sources</td>
<td>Enterprise-grade high-performance messaging middleware, time-series data warehouse and data steaming platform</td>
<td>Set of development tools and libraries for the creation, historical simulation and optimization of quantitative models</td>
<td>High-performance ultra-low latency platform for the real-time live deployment, management and hosting of quantitative models</td>
</tr>
<tr>
<td>Normalize and validate data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintain data archive for subsequent analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stream data to research and real-time models</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TimeBase
- Enterprise-grade high-performance messaging middleware, time-series data warehouse and data steaming platform

### QuantOffice
- Set of development tools and libraries for the creation, historical simulation and optimization of quantitative models
- Convert quantitative ideas to executable models
- Test it on historical data, analyze results, optimize and assess predictive power
- Simulate these models with real-time data

### QuantServer
- High-performance ultra-low latency platform for the real-time live deployment, management and hosting of quantitative models
- Deploy custom and standard quant models in production to generate various deliverables:
  - Alerts, surveillance and risk signals
  - Reports
  - Trades and orders
TimeBase Use Cases

- Time-Series Data analysis
- Algorithm Back-testing
- Live Data streaming / Algorithmic Trading
- Warm-up/Live hybrid
- Market Data Aggregation / Ticker plant

- Messaging Middleware / Streaming Middleware
- Real time event processing
- Message Bus
- Complex Event Processing
- Long distance streaming
Key Features

• **Battle-tested.** 15 years in production with various implementations in the domain of financial markets

• **Cost-efficiency.** Impressive overall TCO and keeps the cost under control as system scales. The minimal fully functional installation is available on a single desktop/cloud instance, linearly scales to a large cluster. Doesn’t require special hardware, learning new language or programming techniques. Automated maintenance and data retention. The initial installation/deployment takes minutes.

• **Superior performance.** Million events per core per second and stable low microsecond latencies on community/cloud hardware.

• **Complete ecosystem.** Production ready integrations with various databases/message brokers/visualization tools (SQL Server, ClickHouse, TimeScale, S3, Kafka, Grafana, PowerBI and others). Connectors for 100+ market data venues.

• **Designed for analytics.** Supports both real-time and historical analytics, seamless transition between historical and live data. Native integration with Jupyter Notebook.

• **Multi-language API.** TimeBase provides API libraries natively implemented on most popular backend languages: Java/C++/C#/Python. Web APIs (REST/WS). Go API is under development.

• **Intelligent Data Modeling.** TimeBase is schema-oriented where the principle of introspection allows using classes in selected languages as schemas. TimeBase schemas allow to represent complex data models (like MBO or ITCH) utilizing features like polymorphic classes, arrays and nested objects. Essential data types are supported including nano-second timestamps, decimal floating-point numbers. Built-in performance-efficient serialization framework.
Key Technical Features

- Rich data schema formats, polymorphism, automatic codec generation.
- Complex Event Processing
- API: TimeBase provides the clients API for Java, C++, C#, and Python programming languages. REST / WebSocket consumer API.
- Queries: TimeBase offers SQL-like query language to retrieve data from single stream
- Support for HDFS / Azure Data Lake / SQLDB as alternative storage layer
- Connectors for 100+ market data venues, Normalized format to represent BBO / MBL / MBO market data.
- Data compression (GZIP) and data encryption (SSL)
- Support for UDP and IPC transport for local clients
- Monitoring solutions for Zabbix, JMX, SNMP
- Replication, Backup, Data Repair
- Deployed using installer, Ansible, or Alpine-based docker image
- Commodity hardware, on-premises or cloud, Linux / Windows
What Makes TimeBase unique

**KEY DIFFERENTIATORS**

- TimeBase offers real-time and historical data using **single** high-performance **streaming API**. Under the hood system may be tuned to stream data with sub-microsecond latencies or read/write millions of messages per second on each data producer and consumer. When streaming live data TimeBase can feed real-time consumers from memory buffers rather than disk (this feature provides significant latency reduction).

- While most of competitors offer key/value sets or BLOB messages, TimeBase natively supports complex **message structures that reflect Business Domain**. For example, TimeBase can store business layer messages that reflect Order Book snapshots or incremental updates, as well as Trade Order requests.
TimeBase Concepts

- TimeBase database has many **Streams**
- Each stream contains many **Messages** of one or several types
- Type of each message is defined by Schema. Each message has
  - Timestamp
  - Symbol
  - Some number of custom fields (simple or composite), according to schema
- Producers use “**Loaders**” (load data into TimeBase)
- Consumers use “**Cursors**” (iterate over data read from TimeBase)
- Stream types:
  - Durable (persistent)
  - Transient (in-memory only)
  - **Topics** (Bypass server)
- TimeBase has **QQL** Query Language to query data or describe/modify stream data schema.
Data Ingestion

• Market Data Connector API (100+ connectors developed)
• Aggregator = Manager for connectors
• Simple API clients (Java, C#, Python)
• Normalized Market Data format (L1/L2/L3)
Data Processing

- Historical Data: Algorithm Back-testing, Quantitative Research: QuantOffice
- Live Market Data: Aggregation / Trading
- Custom API clients (Java/C#/Python/REST/WS)
Performance

• Throughput:
  • TimeBase Streams (TCP)
    • 1 Producer x 1 Consumer = 1.8 M messages/sec
    • 1 Producer x 4 Consumers = 5.5 M messages/sec
  • TimeBase Topics (100 bytes payload)*
    • 1 Producer x 1 Consumer = 9 M messages/sec
    • 1 Producer x 4 Consumers = 24 M messages/sec (6M per consumer)

• Latency (one way, microsecs.):

<table>
<thead>
<tr>
<th>Streams</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td># Mean = 49.2 (StdDev = 120)</td>
<td># Mean = 0.372 (StdDev = 0.186)</td>
</tr>
<tr>
<td>Percentiles:</td>
<td>Percentiles:</td>
</tr>
<tr>
<td>50% = 46</td>
<td>50% = 0.356</td>
</tr>
<tr>
<td>90% = 56</td>
<td>90% = 0.373</td>
</tr>
<tr>
<td>99% = 62</td>
<td>99% = 1.404</td>
</tr>
<tr>
<td>99.9% = 66</td>
<td>99.9% = 1.484</td>
</tr>
<tr>
<td>99.99% = 7065</td>
<td>99.99% = 1.643</td>
</tr>
<tr>
<td>99.999% = 8781</td>
<td>99.999% = 4.735</td>
</tr>
<tr>
<td>Max = 9822</td>
<td>99.9999% = 34.239</td>
</tr>
<tr>
<td></td>
<td>Max = 130.943</td>
</tr>
</tbody>
</table>

*See TimeBase Topics presentation for more information
Hardware / Software Requirements

**Hardware**
- Can run on single laptop or server farm. Commodity hardware
- Back-testing – more RAM is good for cache (minimum 0.5G)
- HDD/SSD/NVME – depends on use case (from 100Mb to 100Tb)
- CPU – depends on use case (minimum 2)

**Software**
- Windows 10 / Windows Server 2019
- Linux (CentOS, RHEL, Amazon Linux, Ubuntu)

**Cloud**
- Ansible
- Docker / Docker-compose
- Kubernetes