



TimeBase

Technical Presentation

Overview

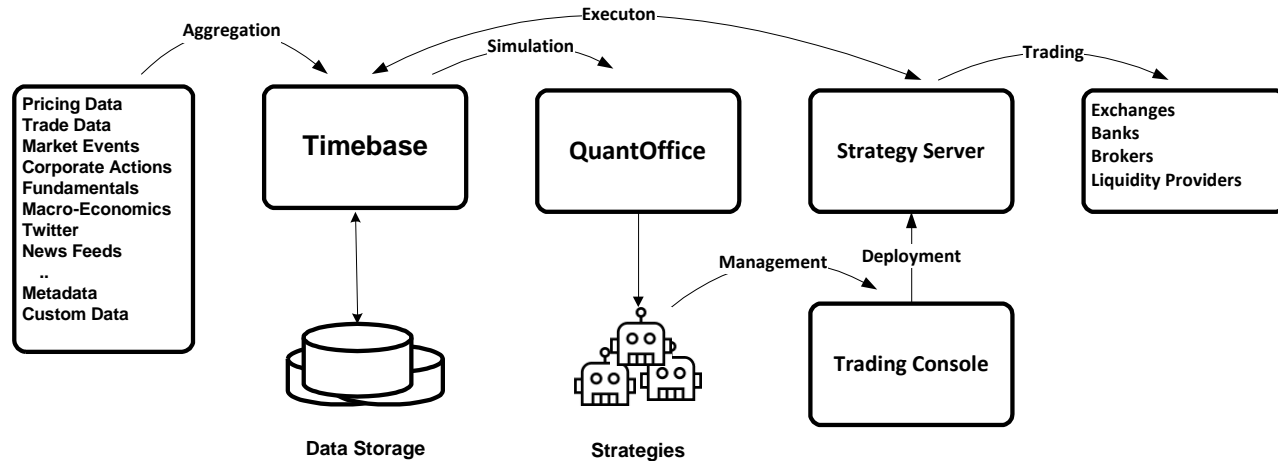
- **TimeBase** is a high-performance *time series* database and *streaming system* developed by EPAM Real Time Computing Lab.
- Current TimeBase is a result of 15 years of experience in financial domain. TimeBase runs standalone or in a cluster, processes millions of messages per second, stores terabytes of data, and can offer sub-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, Kuber)
- 2020 – TimeBase Community Edition (timebase.info)

TimeBase Use Cases

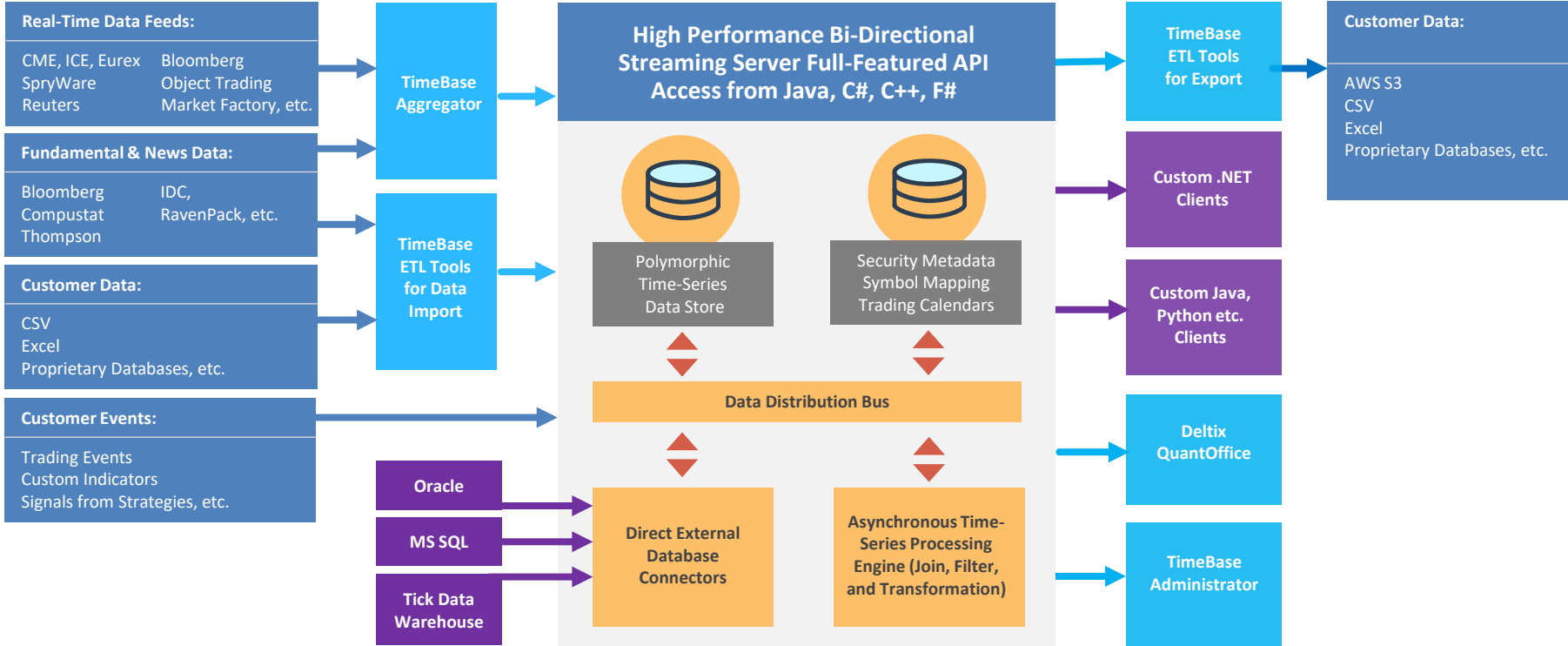
- Time-Series Data analysis
- Algorithm Back-testing
- Live Data streaming / Algorithmic Trading
- Warm-up/Live hybrid
- Market Data Aggregation / Ticker plant
- Messaging Middleware / Streaming Middleware
- Real time event processing
- Message Bus
- Complex Event Processing
- Long distance streaming



Key Features

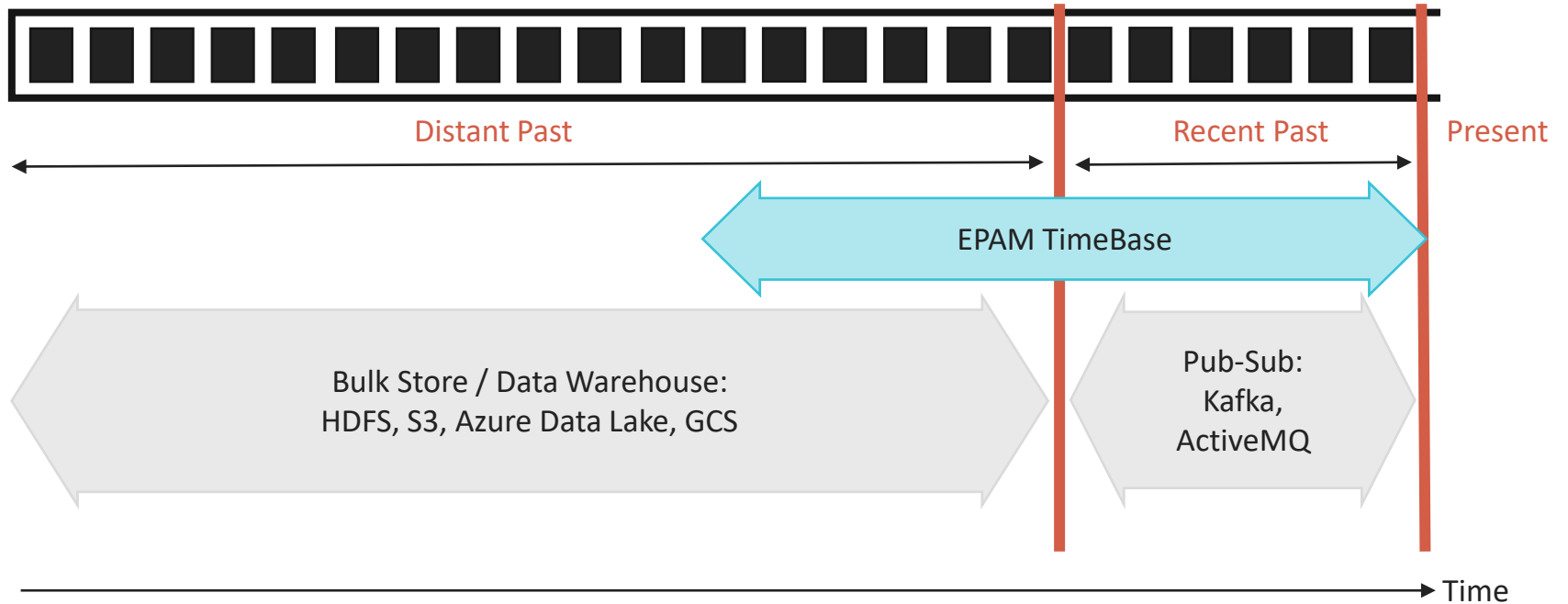
- Rich data schema formats, polymorphism, automatic codec generation.
- Complex Event Processing
- API: Timebase provides the clients API for Java, C++, C#, and Python programming languages. REST / WebSocket consumer API.
- Queries: Timebase offers SQL-like query language to retrieve data from single 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

Architecture Diagram



Real Time and Historic Data Processing Landscape

Anatomy of Data Stream



How Timebase stacks up against COMPETITORS

Features & Functions	Timebase	KDB	OneTick	InfluxDB	ActiveMQ	Kafka	Timescale
Historical Data sets	●	●	●	●	○	●	●
Live Data Streaming	●	●	●	◐	●	●	○
Query language	◐	●	●	●	○	◐	●
Availability of Developer Resources	●	○	○	●	●	●	●
Historical data modification	●	●		○	○	○	●
Historical data deletion	●	●		◐	○	◐	●
Timestamp resolution	●	●	●	○	○	◐	○
Complex message formats	●	◐	●	◐	◐	◐	◐
Data format evolution/versioning	●	○	●	○	○	○	●
Per-stream performance tuning	●	○	●	○	○	●	◐
Live streaming latency	●	●	●	◐	◐	○	○
Streaming throughput 1P1C	●	●	●	◐	○	◐	◐
Cost	\$	\$\$\$	\$\$\$	Enterprise features are \$\$\$		Freemium	\$\$

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.

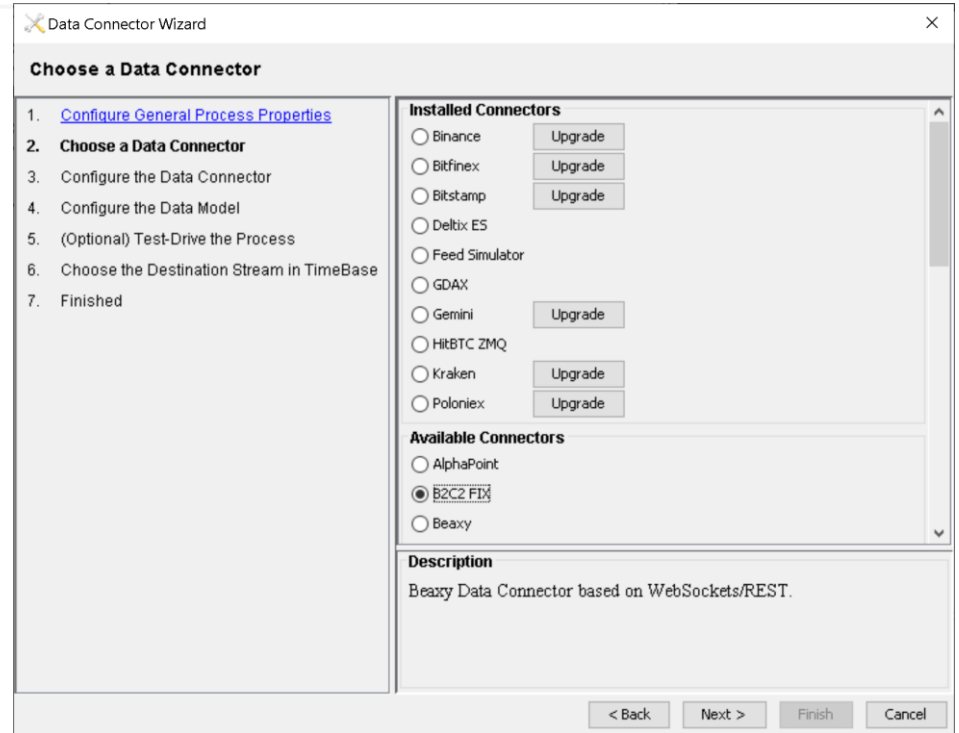
TimeBase Concepts

- 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* (consumer-to-consumer, bypass server)
- Timebase has **SQL** 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)



Timebase is integrated with 100+ Market Data Venues

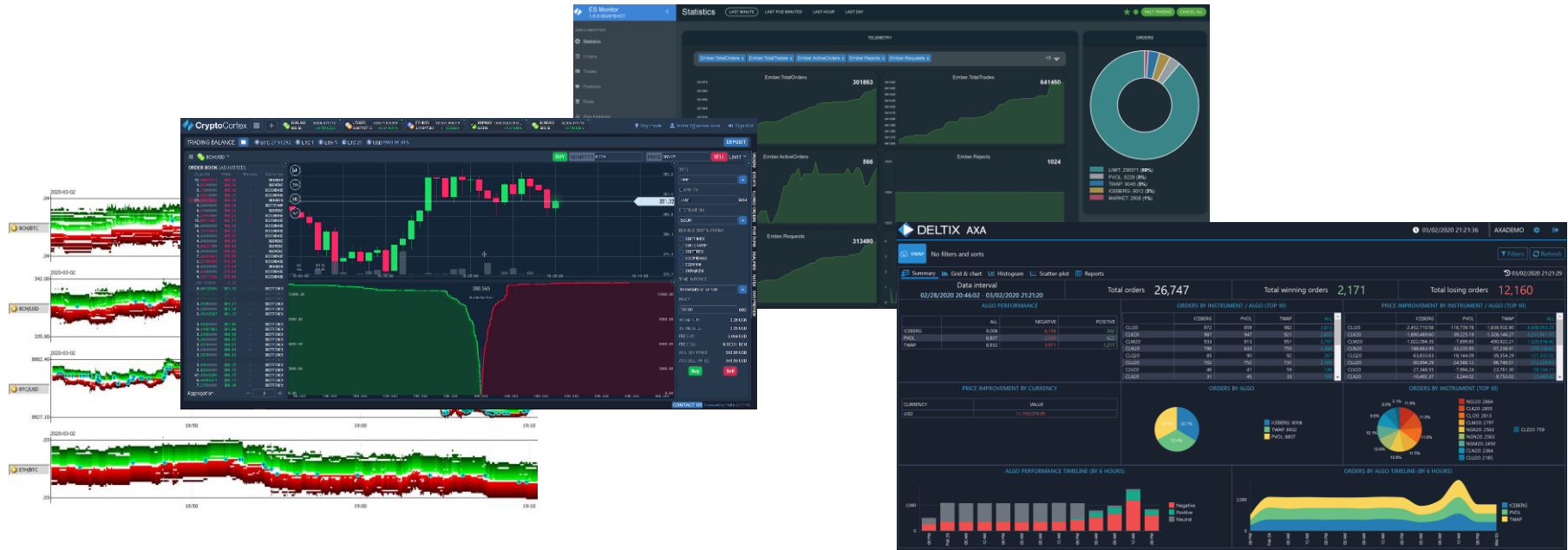
Very short time to market for new integrations.

Dedicated team of Software Engineers and QA Engineers who have 10+ years of experience with Market Data APIs



Data Processing

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



Security

- UAC – user authentication and authorization
 - Users, Groups (LDAP/ActiveDirectory)
 - Permission Rules: { ALLOW/DENY, Principal, Operation, Resource }*
- DAC – data access control [per-message]
 - Implemented for a few data vendors/clients to meet data distribution requirements (Bloomberg, Reuters)
- SSL – data in transit encryption (Considerable CPU penalty, recommended for WAN clients only)
- SSO (Timebase Admin web GUI)
- Development follows OWASP top 10 standard:
 - Spotbugs (with security plugin)
 - Dependency Check (scan of third-party dependencies against OWASP vulnerabilities database)

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, microseconds):

Streams

Mean = 49.2 (StdDev = 120)

Percentiles:

50% = 46

90% = 56

99% = 62

99.9% = 66

99.99% = 7065

99.999% = 8781

Max = 9822

Topics

Mean = 0.372 (StdDev = 0.186)

Percentiles:

50% = 0.356

90% = 0.373

99% = 1.404

99.9% = 1.484

99.99% = 1.643

99.999% = 4.735

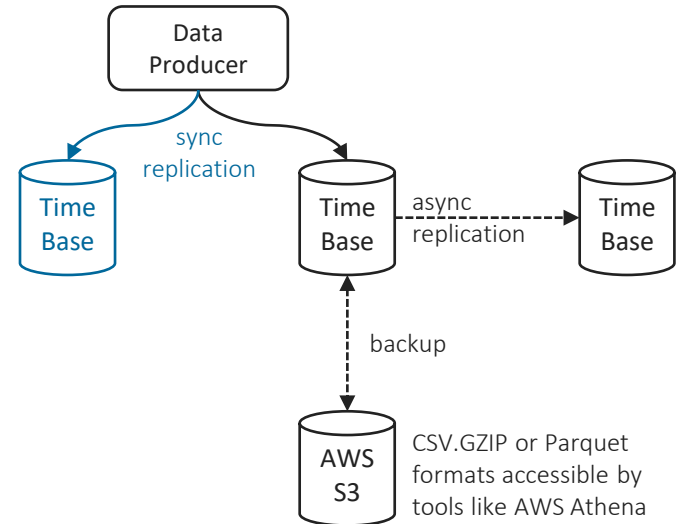
99.9999% = 34.239

Max = 130.943

See TimeBase Topics presentation for more information

TimeBase Failover and Disaster Recovery Options

- Asynchronous Stream Replication
 - Can be configured at each stream level
 - Can filter instruments/message types
 - Can run in live (real-time) or in batch (periodic) mode
- Synchronous Stream Replication:
 - Producer writes to multiple TimeBase instances (API)*
- Disaster Recovery
 - Backup / Restore to “cheap” AWS S3 storage in batch/live mode
 - Asynchronous replication to TimeBase running in another data center
 - TimeBase has “Repair Shop” tool



Open Source Components

- Timebase is in-house development of Deltix, Inc (acquired by EPAM February 2020).
- Notable dependency on FOSS software:
 - [Aeron](#) messaging library from Real Logic is used as transport layer for TimeBase Topics
 - [Apache Tomcat](#) is used for HTTP/TCP server, can be replaced by Jetty or similar.
- Only non-restrictive licenses (Apache License 2.0, LGPL, etc.)

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)

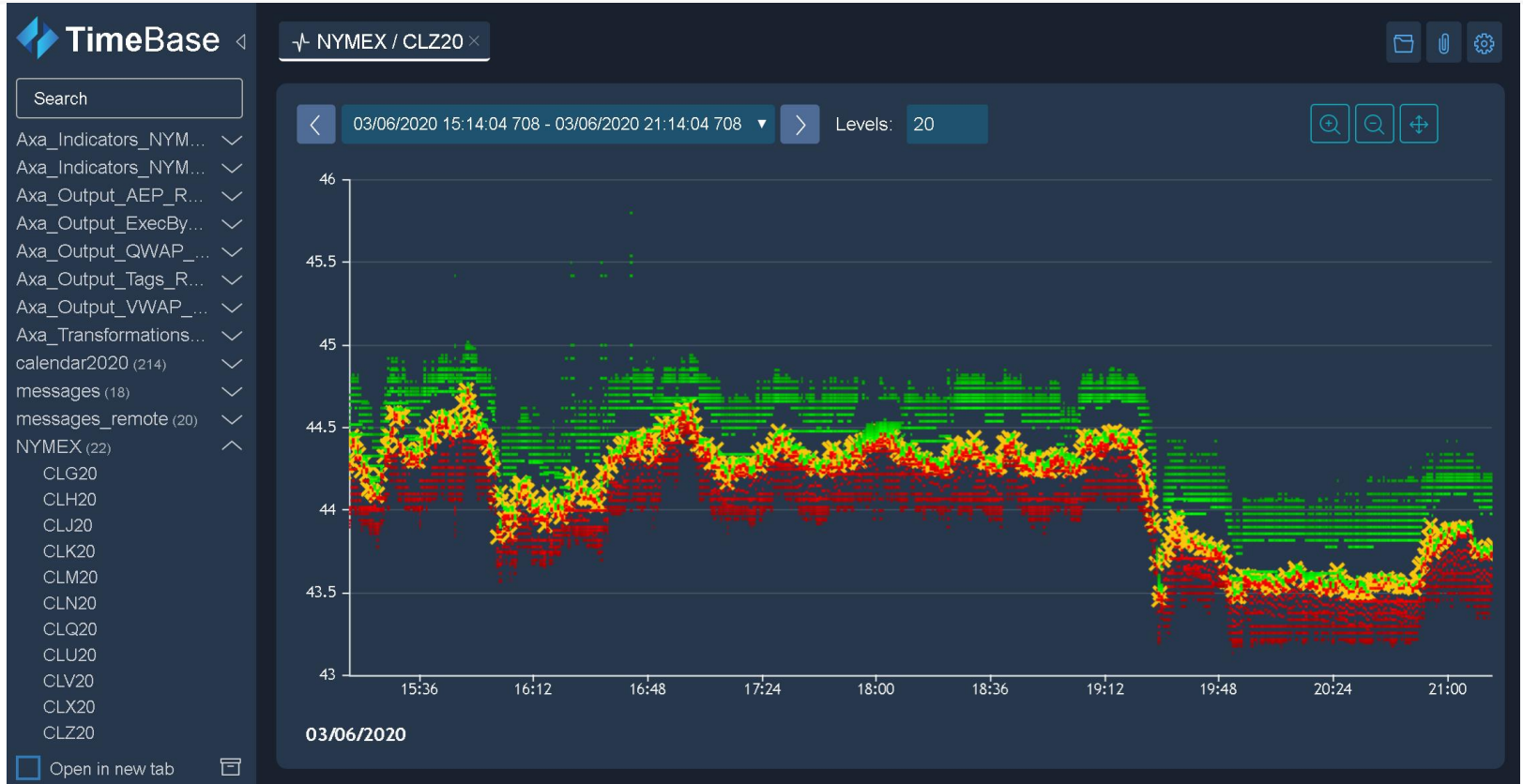
CLOUD

- Ansible
- Docker / Docker-compose
- Kubernetes

SOFTWARE

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

TimeBase Admin – web client



TimeBase Admin – desktop client (Java)

The screenshot displays the TimeBase Administrator v5.3.49 desktop application. The window title is "TimeBase Administrator v5.3.49 - SNAPSHOT-?-0e7e121020 - admin@localhost:8011". The interface includes a menu bar (File, View, Data, Tools), a toolbar, and a database browser on the left showing a tree view of databases under "TimeBase", including "COINBASE" and "Coinbase Data Connector".

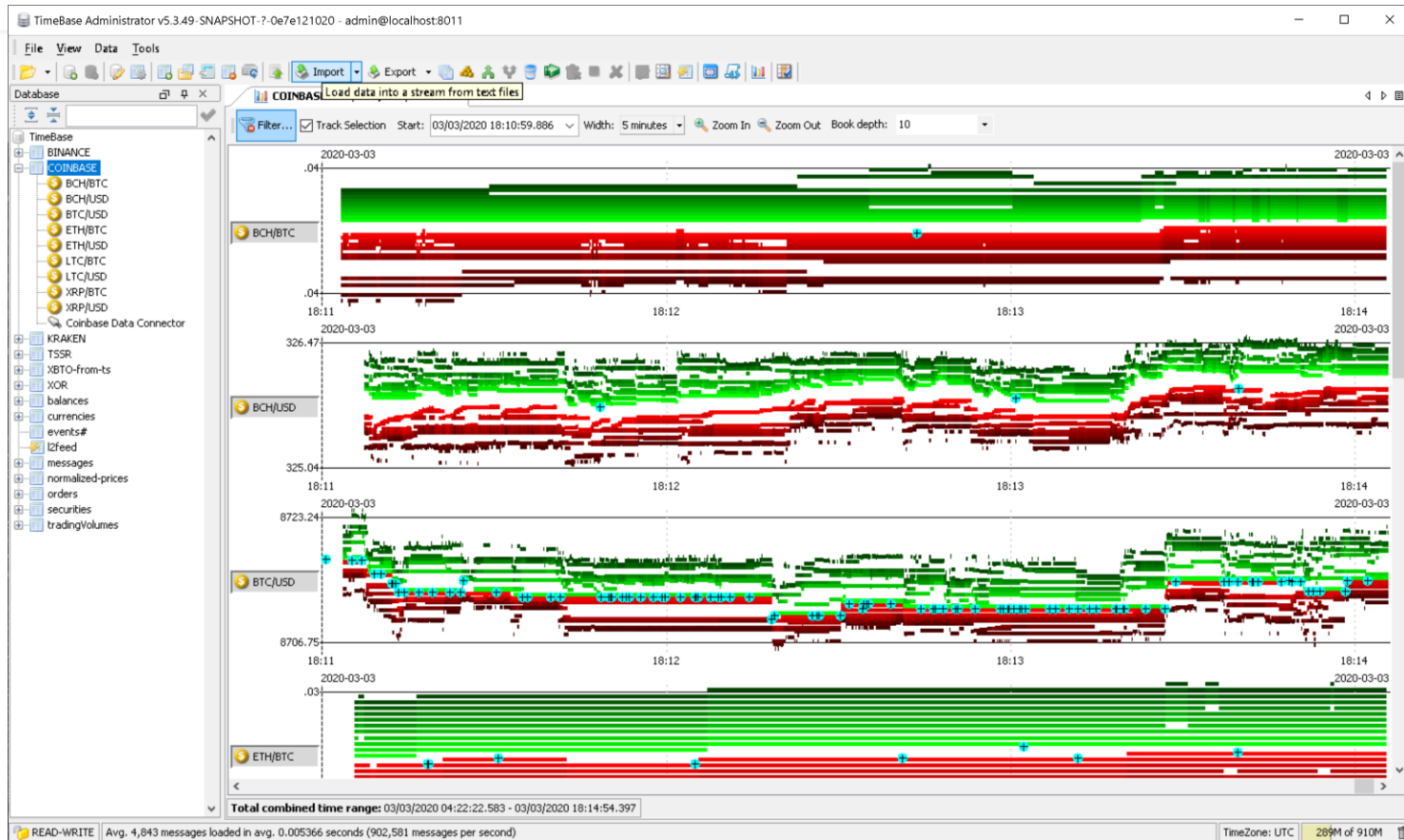
The "Query Editor" window shows the following SQL query:

```
1 select *
2 from COINBASE
3 where symbol='BTC/USD' and timestamp > '2020-03-03 17:21:40'd
```

The "Execute" button is visible, along with a "Max rows: 100" setting. The main display area shows a table with the following columns: Instrument..., Symbol, Identifier, Time, Raw Times..., Market Message (Currency..., Original Ti..., Sequence..., Source Id), Security Status Change Message (Cause, Exchange..., Original St..., Status), Package H..., and Coinbase P... (Vendor SN..., Package T..., Entries). The table contains 990 rows of data, with the first few rows showing instrument symbols like KRANKEN, TSSR, XBTO-from-ts, XOR, and various balances and messages for BTC/USD.

At the bottom of the window, a status bar indicates "100 rows loaded in 0.101000 seconds (990 rows per second)" and "Avg. 4,843 messages loaded in avg. 0.005366 seconds (902,581 messages per second)". The system tray shows "TimeZone: UTC" and "300M of 910M".

TimeBase Admin – desktop client (Java)



QuantOffice charting for TimeBase streams

