Member-only story

Timeplus Proton: The Lightning-Fast SQL Engine for Real-Time Streams

4 min read · 2 days ago

Algo Insights Following

Listen Share More

Real-time data is a mess. If you've ever tried building a streaming pipeline, you know the chaos. Kafka for streams, Flink or ksqlDB for processing, ClickHouse for analytics, connectors for… everything else.

Before long, you're knee-deep in YAML and microservices, wondering if this was meant to be *fun*.

Then I discovered **Timeplus Proton**. It's essentially: *"What if stream processing was just SQL — and it ran like a rocket?"*

Fastest SQL pipeline engine for stream processing, analytics, observability and AI

· · ·

So, What's Proton?

Proton is a single binary — written in C++ — that performs stream processing, analytics, observability, and even a touch of AI magic.

No JVM.
No ZooKeeper.
No massive cluster setup.

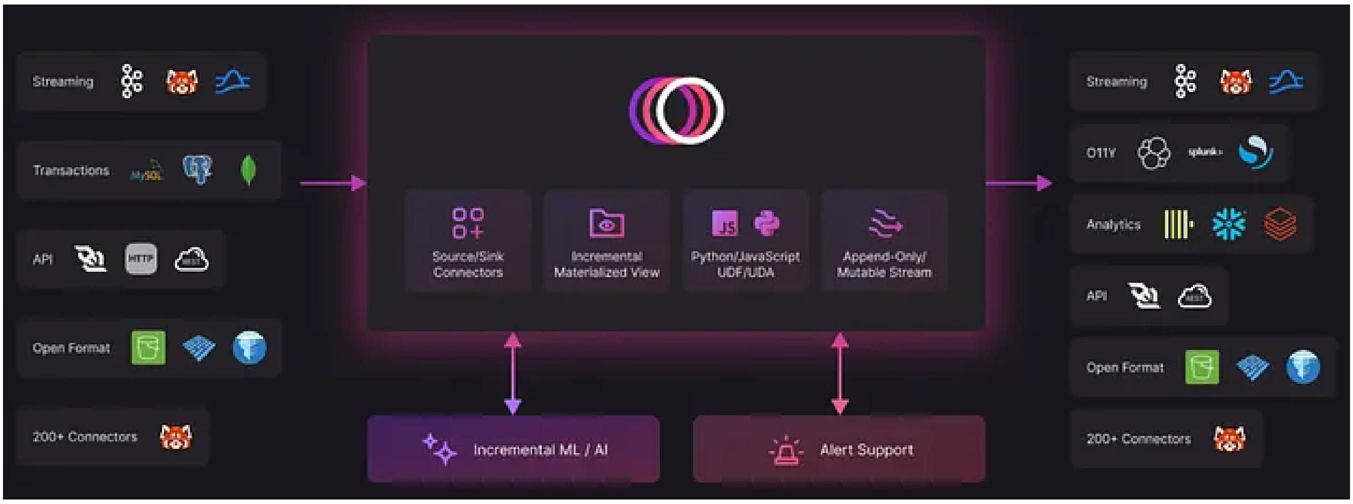
You just run:

```
curl https://install.timeplus.com/oss | sh
```

And you're live. Literally.

Behind the scenes, it uses the **ClickHouse engine**, which is already one of the fastest databases available. However, Proton adds all the streaming features that ClickHouse doesn't offer — joins, windows, materialised views, watermarks, and more.

It's like ClickHouse has learned to listen to data as it occurs.



• • •

The Cool Part — It's Actually Fast

I'm not talking "marketing fast."
I mean, "90 million events per second on a MacBook" fast.

Proton's written in C++ and uses SIMD optimisation (which basically means it squeezes more work out of your CPU).
And because it's a single binary under 500MB, it doesn't chew through memory or spin up ten background services to say "hello."

You can even run it on an AWS t2.nano — that's one vCPU and 0.5GB memory. Try doing that with Flink.

. . .

ClickHouse vs Proton (in simple terms)

Here's how I see it:

- **ClickHouse** is for *big historical analytics*. It's a data warehouse.
- **Proton** is for *real-time decisions*. It's a stream engine.

ClickHouse can read from Kafka, sure, but it doesn't *think in streams*. It thinks in tables. Proton lives in both worlds — it unifies real-time and historical data. You can detect something now and query it later, without juggling two systems.

It's similar to ClickHouse and Flink merging into a small binary.

. . .

Getting Hands-On

Once installed, open Proton's SQL client and begin writing SQL—no new language to learn.
Example — reading from Kafka and writing to ClickHouse:

```
CREATE EXTERNAL STREAM aws_msk_stream (
  device string,
  temperature float
)
SETTINGS
  type='kafka',
  brokers='prefix.kafka.us-west-2.amazonaws.com:9098',
  topic='topic',
  security_protocol='SASL_SSL',
  sasl_mechanism='AWS_MSK_IAM';
```

Then send the results to ClickHouse:

```
CREATE EXTERNAL TABLE ch_aiven
SETTINGS type='clickhouse',
  address='abc.aivencloud.com:28851',
  user='avnadmin',
  password='..',
  secure=true,
  table='events';
```

And finally, create a view that aggregates data every 10 seconds:

```
CREATE MATERIALIZED VIEW mv_msk2ch INTO ch_aiven AS
SELECT window_start AS timestamp, device, avg(temperature) AS avg_temperature
FROM tumble(aws_msk_stream, 10s)
GROUP BY window_start, device;
```

That’s real-time ETL — no frameworks, no YAML, just SQL.

. . .

Don’t Have Kafka? No Problem.

You can generate random data for testing.

```
CREATE RANDOM STREAM devices(
  device string default 'device'||to_string(rand()%4),
  temperature float default rand()%1000/10);

SELECT device, count(*), min(temperature), max(temperature)
FROM devices GROUP BY device;
```

You’ll see a live stream of fake IoT data passing by. It’s oddly satisfying.

```
┌─device───┬─count()─┬─min(temperature)─┬─max(temperature)─┐
│ device0 │ 2256 │ 0 │ 99.6 │
│ device1 │ 2260 │ 0.1 │ 99.7 │
│ device3 │ 2259 │ 0.3 │ 99.9 │
│ device2 │ 2225 │ 0.2 │ 99.8 │
└──────────┴─────────┴──────────────────┴──────────────────┘
```

Feels like magic.

. . .

Why It Matters

You can use Proton for:

- Real-time dashboards
- Streaming ETL and data prep
- Alerts and anomaly detection
- IoT and sensor data
- Log analytics and observability

It replaces a *stack* of tools with a single binary and a single language: SQL.

You can deploy it anywhere — Docker, Homebrew, cloud — or simply run it locally to experiment.

```
docker run -d -p 8123:8123 -p 8463:8463 \
  --name proton d.timeplus.com/timeplus-io/proton:latest
```

. . .

My Take

Proton seems like a tool you try out of curiosity… and then realize it quietly replaces three things in your stack.

It's fast. It's clean. It doesn't yell at you for not setting a JVM flag, and it speaks the most universal language in data: SQL.

If you work with live data streams — or have ever wished Flink were easier — give this a try:

```
curl https://install.timeplus.com/oss | sh
```

You'll have a streaming SQL engine up and running before your coffee cools.

Sql Sql Server Live Streaming AI Data

Following

Written by Algo Insights

3.4K followers · 6 following

Algo Insights: Unlocking the Future of Trading with Code, Data, and Intelligence. https://linktr.ee/Algo_Insights

No responses yet

Bgerby

What are your thoughts?

More from Algo Insights

In Coding Nexus by Algo Insights

4 Open-Source Tools That Made Me Rethink My Dev Setup

I’ve been coding for a while now. Most of the tools we use every day… they’ve been the same for years. Editors, browsers, frameworks. Just…

Sep 3 450 9

In Coding Nexus by Algo Insights

How to Build Your Own AI Rig for Running Local LLMs (Gemma, Mistral, Qwen, GPT-OSS and Llama)

About three months ago, I realised I was utterly dependent on companies that didn't care about anything except power, Money, and control.

✦ Oct 8 👏 354 💬 12

In Coding Nexus by Algo Insights

DSPy: Stop Prompting, Start Programming Your AI

If you've ever spent hours tweaking prompts, you know how frustrating it can be.

✦ Oct 1 248 6

Algo Insights

5 Prompting Secrets AI Engineers Use (That Most People Don’t Know)

Most folks treat AI like Google. They type in a question, hit enter, and hope something useful comes back.

✦ Sep 25 376 10

See all from Algo Insights

Recommended from Medium

In Meyka Finance Wire by Huzaifa Zahoor

How to Build a Stock Screener with Custom Technical Indicators

When a client needed to track 900 stocks with specific technical indicators, standard screening tools could not do the job. We built a...

✦ 6d ago 👏 87 💬 3

In Level Up Coding by Fareed Khan

Building an Agentic Deep-Thinking RAG Pipeline to Solve Complex Queries

Planning, Retrieval, Reflection, Critique, Synthesis and more

4d ago 801 4

In CodeToDeploy by TechToFit - Master Your Life with Tech

I Tried Google’s New AI Agents. It’s a Gold Rush.

I spend my days deep in the world of AI, but every so often, something drops that makes me stop everything. This is one of those times…

Oct 10 164 3

Algo Insights

10 AI Websites You Can’t Ignore in 2025

AI has sped up this year, reminding me of times I’d spend hours pounding on a keyboard to fix a bug.

✦ 6d ago 13 1

In Towards Deep Learning by Sumit Pandey

Why Everyone Will Want DGX Spark on Their Desk — Yes, Everyone

I just saw this picture today and was amazed, I’ve been waiting for this moment for a long time. (No, it’s not Elon.) It’s that tiny...

✦ Oct 14 23 7

In Coding Nexus by Civil Learning

MarkItDown: Convert Anything into Markdown—the Smart Way to Feed LLMs

You know that feeling when you're trying to feed a PDF or a Word document into an LLM, and it just doesn't understand what's going on…

✦ Oct 15 👏 144 💬 3

See more recommendations