Grid Dynamics

The Streaming Data Platform

as a heavy duty enterprise data marketplace

when to use it in your project

Hello

Denis Golovachev

Grid Dynamics

SPB

Saint-Petersburg


BIG
DATA

Software Architect

Data Streaming

The Streaming Data Platform as a heavy duty enterprise data marketplace When to use it in your project. The case of MMA

The Streaming Data Platform
as a heavy duty enterprise data marketplace
When to use it in your project.
The case of MMA

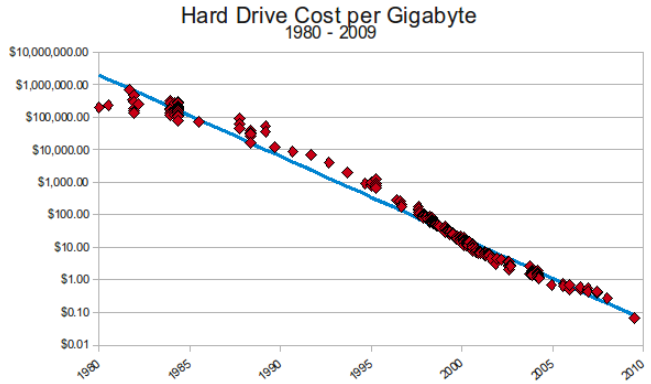
The Streaming Data Platform
as a heavy duty enterprise data marketplace
When to use it in your project.
The case of MMA

MMA - Multichannel Marketing Automation

Small Introduction to

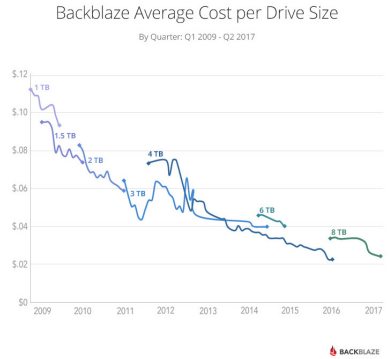
Streaming Platforms

Let’s begin
Denis, Architect, engineer, more tech details in this part
Begin with small introduction. Some may have attend, but I’d like to present slightly different view / manner
And as I’m a tech
digits and plots



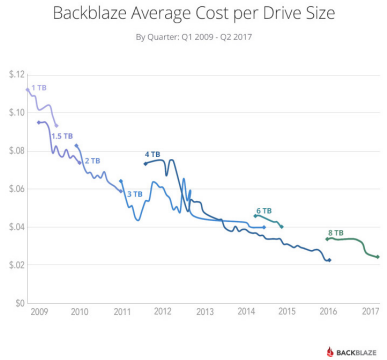
For the past 40+ years hard drives prices are constantly dropping

In our electronic devices we usually have HDDs and SSDs as storage
This plot shows how how price changed over the last 40 years
Constantly dropping
Twice a year
From around $500,000 per gigabyte in 1981 to less than $0.02 per gigabyte today



- 2010 - countries could store everything. i.e. China
- 2015 - big companies could afford to store everything. i.e. Google
- 2019 - everyone!

So it became possible somewhere around 2010 for contries to store every data they could collect/intercept/sneak
Sometimes for their survailance programs or even just in case
Somewhere in 2015 it become possible for large companies to collect everything

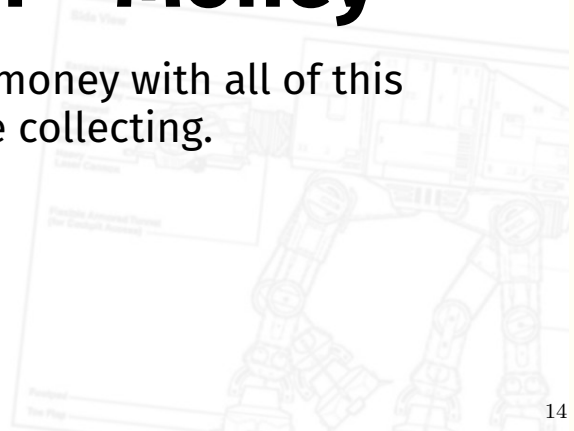


So, the price is low, we store everything.

And i.e. if we're Facebook it's a question whether we should create a task for our IT guys to clean up outdated data,
And you know that IT time is expensive. So maybe it's better just to buy one more drive for storage.
well this task could take several hours and Facebook IT time is expensive
More than several cents

Information = Money

But could we earn **more** money with all of this information we collecting.



That’s what basically we do in our BigData field

And we’re good at it

The world’s most valuable resource is no longer oil, but data

So nowdays everyone says that
the world’s most valuable resource is no longer oil,
but data

- Cheap Storage
- Collect Everything
- Extract Values
- … Profit

Time to make profit
But sometimes it’s not enough to collect lot of data and extract value from it. Let me explain with example

Do you know this company?

The Knight Capital Group was an American global financial services firm engaging in market making, institutional sales and trading.
With its high-frequency trading algorithms **Knight was the largest trader in U.S**.

Knight

- Lot of Events/Data
- Lot of Analytics

Lot of data - Events, Signals, Markers

On August 1st, 2012, Knight Capital deployed a new software update to their production servers. They switched it on and immediately they started losing literally $10 million [£6.4m] a minute.

Once they deployed a new software update to their production servers
They started losing literally $10 million a minute
And this went on for 45 minutes.
At the end of it all they wound up having lost $440 million

Humans still **watch the systems**, but **the computers move far too quickly for us to react to everything they do**.

In a postmortem they said..
Due to computer glitch the company was making trades it didn't intend to make.
That's guys an example how to lose almost half a billion dollars in a little over half an hour.
It was a fatal day for them

The ability of understanding, processing and monitoring huge amount of data fast could save you life!

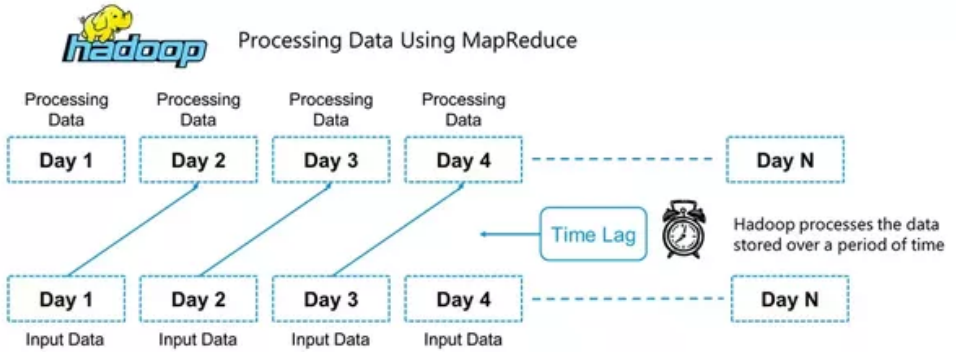
conclusion is that sometimes...
Another valuable example is my previous project

My previous project

it was connected with fraudsters, we were fighting against them

Fraud Detection System

We had a lot of events. Different kind of them.
Events from our users and with machine learning we were looking for fraudlent behaviour
Anomalies



- Collect - 24hrs
- Process - 6hrs
- Block Fraudsters - 10 minutes

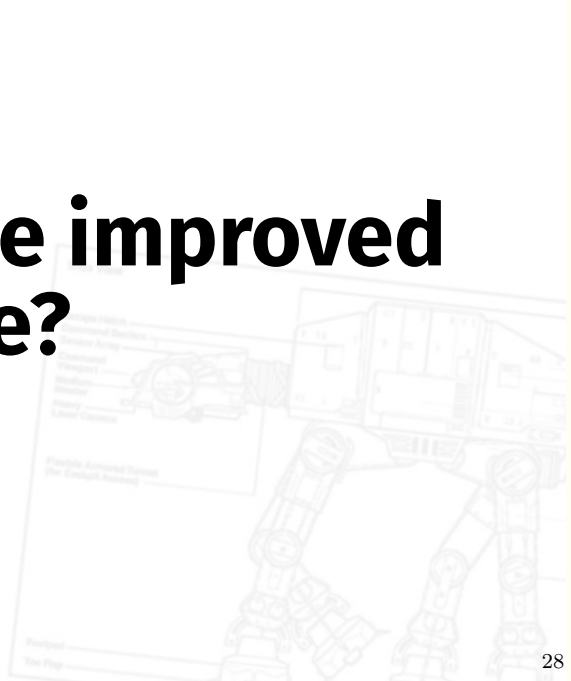
Here is First version of Fraud Detection application design
We were collecting 24h of data, processing it for 6 hours and than report some bad guys

We’re loosing money!

30hr lag →**70 000$ per day**

But once, analytcs department found out that we’re loosing money. Quite a lot!
Fraudsters adapted to change their accounts every day, IPs every day, browsers every day.
And this strategy works for them

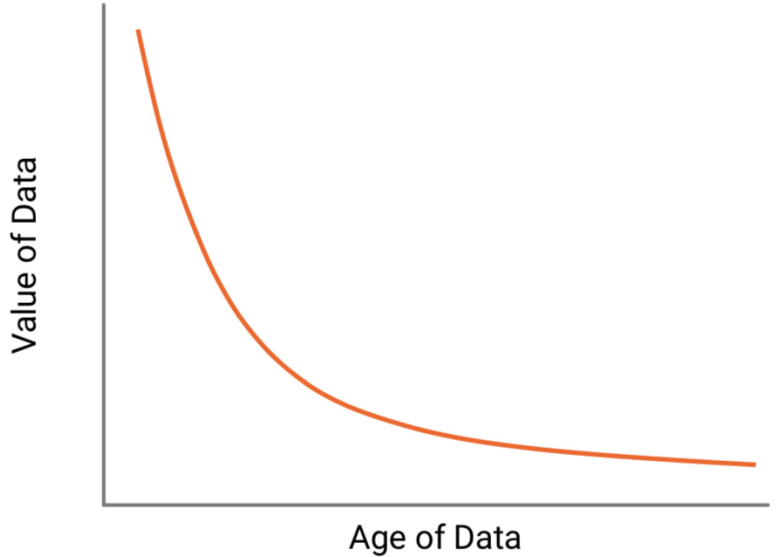
What could be improved here?



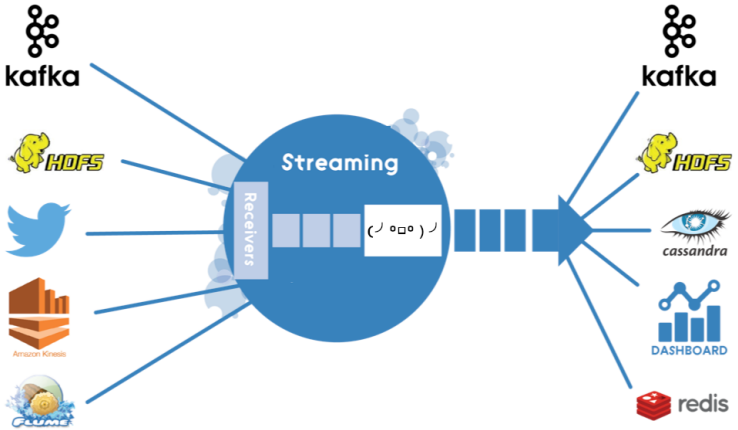
Two examples have quite similar flow

Reaction Time!

Sometimes it’s too late to react



We could collect lot of data but this data may be useless tomorrow.
Even in 1 hour it may be useless for some sorts of analysis
So we decided to try streaming and



30 hrs →10 mins
And Fraudsters were Disappointed

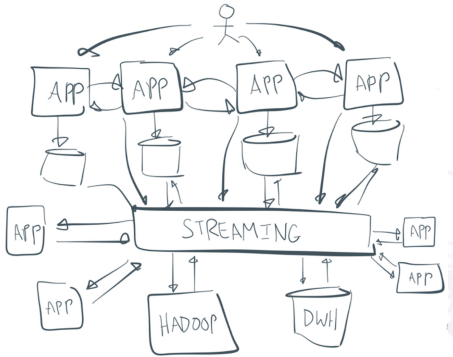
We made it but Knight Capital was not so lucky.
So in general with this kind of streaming solutions
we could <next slide>

- Reduce reaction time
- Minimize risk surface
- Compete for best offers in market
- Give out customers what they need right in time
- ...

Especially for outstanding events
Predict some disasters

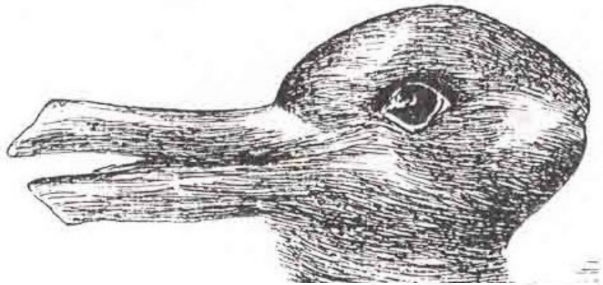
Streaming Data Platform

And we’re building this kind of platform in Aduno,
Streaming Data Platform, SDP project
We’re building capability to be fast
Roughly speaking it looks like this



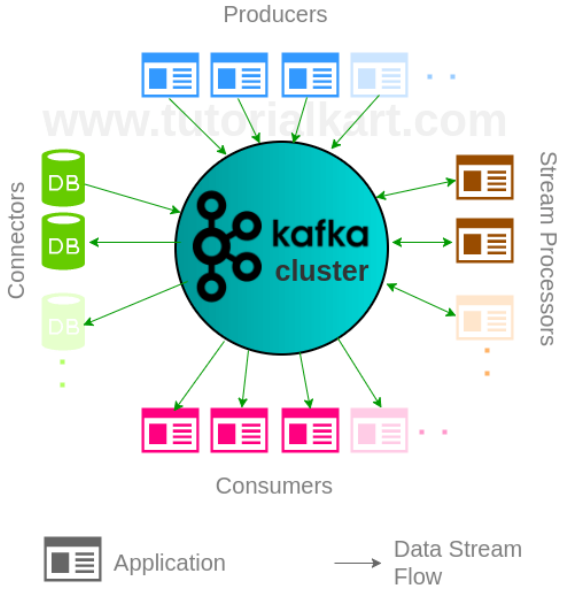
Streaming Data Platform

You may say: But we already have something similar.
Looks like… wait, Denis
Another magic boxes in the middle

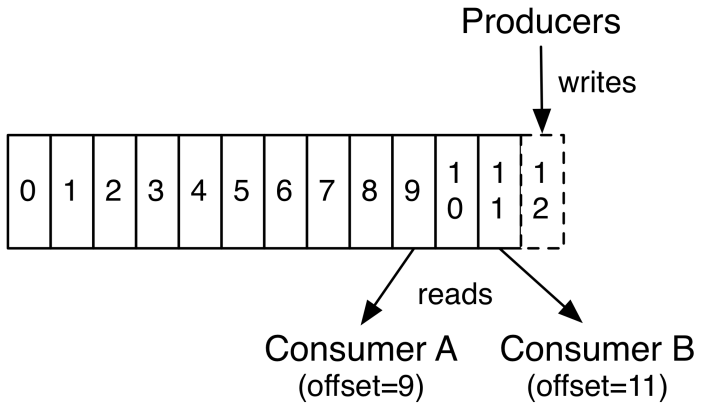


Yet another ESB!?

I could explain, but first I should explain some concepts.
First of all let’s take a look at our middleware



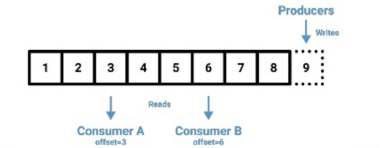
Kaka - it’s database, special case database. Log Storage
Multiple consumers and producers working with immutable log



Immutable event log

Consumers could shift backward and forward along this immutable log

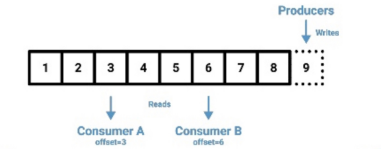
Event Streaming Paradigm



- Don’t route or just processing, it’s storage and processing.

We could not only consume data in motion, but also use it as database
Built-in scalability - you don’t need another separate broker/cluster for another project
Consumers could read on their own pace
But don’t think that SDP and ESBs are enemies, no. they are friends!
They are complementary

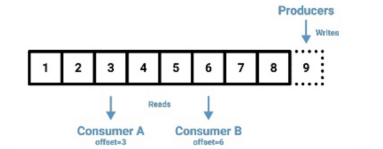
Event Streaming Paradigm



- Don’t route or just processing, it’s storage and processing.
- Allows downtime for maintenance

We could not only consume data in motion, but also use it as database
Built-in scalability - you don’t need another separate broker/cluster for another project
Consumers could read on their own pace
But don’t think that SDP and ESBs are enemies, no. they are friends!
They are complementary

Event Streaming Paradigm



- Don’t route or just processing, it’s storage and processing.
- Allows downtime for maintenance
- Backpressure
- Built-in scalability

We could not only consume data in motion, but also use it as database
Built-in scalability - you don’t need another separate broker/cluster for another project
Consumers could read on their own pace
But don’t think that SDP and ESBs are enemies, no. they are friends!
They are complementary

TIBCO

Informatica

IBM

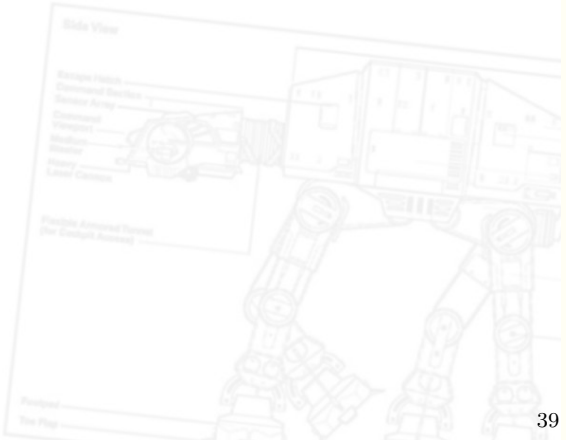
solace

RabbitMQ

ActiveMQ

SDP is scalable, reliable, **but**:

- Integration with legacy



SDP is not silver bullet
But still there are areas in which we can help.

TIBCO

Informatica

IBM

solace

RabbitMQ

ActiveMQ

SDP is scalable, reliable, **but**:

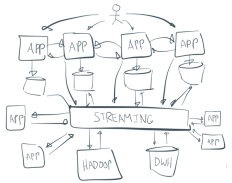
- Integration with legacy
- Fast message delivery [Less than a second]

SDP is not silver bullet
But still there are areas in which we can help.

SDP is scalable, reliable, **but**:

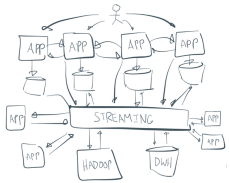
- Integration with legacy
- Fast message delivery [Less than a second]
- Synchronous Point to Point messaging
- Complex Routing
- Default protocol is proprietary

SDP is not silver bullet
But still there are areas in which we can help.



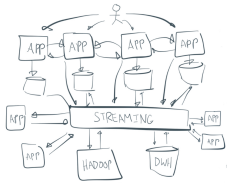
- You have data with short expiration time

Data debt - you don’t want to find yourself in situation when amount of data you collected is so huge that it becomes hard to analyze.
Complience burden - Security, audit … we’re ready for this.
platform with built-in security
Battaries - reprocessing, zero-dataloss maintenances, backpressure
So if in you project you have this things to solve you should consider SDP
We’re always open for your questions and ideas! ty



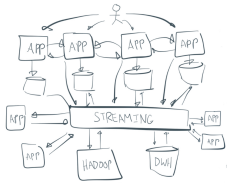
- You have data with short expiration time
- You want to reduce data debt

Data debt - you don’t want to find yourself in situation when amount of data you collected is so huge that it becomes hard to analyze.
Complience burden - Security, audit … we’re ready for this.
platform with built-in security
Battaries - reprocessing, zero-dataloss maintenances, backpressure
So if in you project you have this things to solve you should consider SDP
We’re always open for your questions and ideas! ty



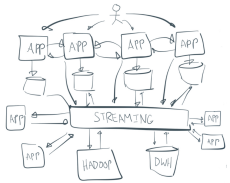
- You have data with short expiration time
- You want to reduce data debt
- Reaction speed is important for you. Especially for outstanding events

Data debt - you don’t want to find yourself in situation when amount of data you collected is so huge that it becomes hard to analyze.
Complience burden - Security, audit … we’re ready for this.
platform with built-in security
Battaries - reprocessing, zero-dataloss maintenances, backpressure
So if in you project you have this things to solve you should consider SDP
We’re always open for your questions and ideas! ty



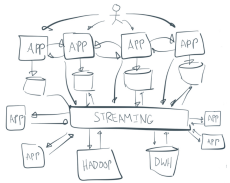
- You have data with short expiration time
- You want to reduce data debt
- Reaction speed is important for you. Especially for outstanding events
- You'd like to build reliable integration between systems. Integration with batteries built-in

Data debt - you don't want to find yourself in situation when amount of data you collected is so huge that it becomes hard to analyze.
Complience burden - Security, audit … we're ready for this.
platform with built-in security
Battaries - reprocessing, zero-dataloss maintenances, backpressure
So if in you project you have this things to solve you should consider SDP
We're always open for your questions and ideas! ty



- You have data with short expiration time
- You want to reduce data debt
- Reaction speed is important for you. Especially for outstanding events
- You'd like to build reliable integration between systems. Integration with batteries built-in
- You think about how to share your data with other projects

Data debt - you don't want to find yourself in situation when amount of data you collected is so huge that it becomes hard to analyze.
Complience burden - Security, audit … we're ready for this.
platform with built-in security
Battaries - reprocessing, zero-dataloss maintenances, backpressure
So if in you project you have this things to solve you should consider SDP
We're always open for your questions and ideas! ty



- You have data with short expiration time
- You want to reduce data debt
- Reaction speed is important for you. Especially for outstanding events
- You'd like to build reliable integration between systems. Integration with batteries built-in
- You think about how to share your data with other projects
- You tired of compliance burden

Data debt - you don't want to find yourself in situation when amount of data you collected is so huge that it becomes hard to analyze.
Complience burden - Security, audit … we're ready for this.
platform with built-in security
Battaries - reprocessing, zero-dataloss maintenances, backpressure
So if in you project you have this things to solve you should consider SDP
We're always open for your questions and ideas! ty

Thank you!

References:

- https://en.wikipedia.org/wiki/Knight_Capital_Group
- https://www.bbc.com/news/magazine-19214294
- https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte/
- https://www.slideshare.net/KaiWaehner/apache-kafka-vs-integration-middleware-mq-etl-esb