

WiFi: HD-Events/znsqv2

How to Use Java Streams to Access Existing Data With Ultra-Low Latency

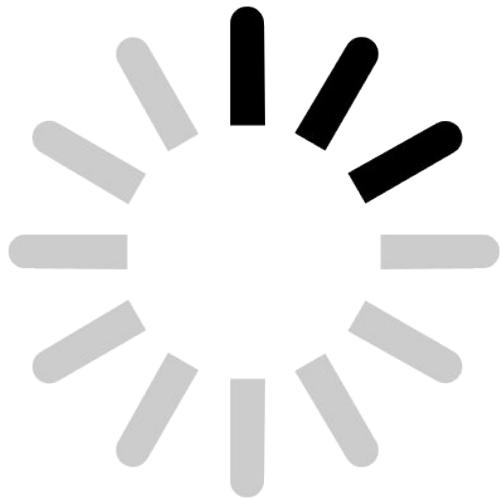
Data Riders Meetup, March 21 2018

Carina Dreifeldt, CEO
Per Minborg, CTO



NEED FOR SPEED™

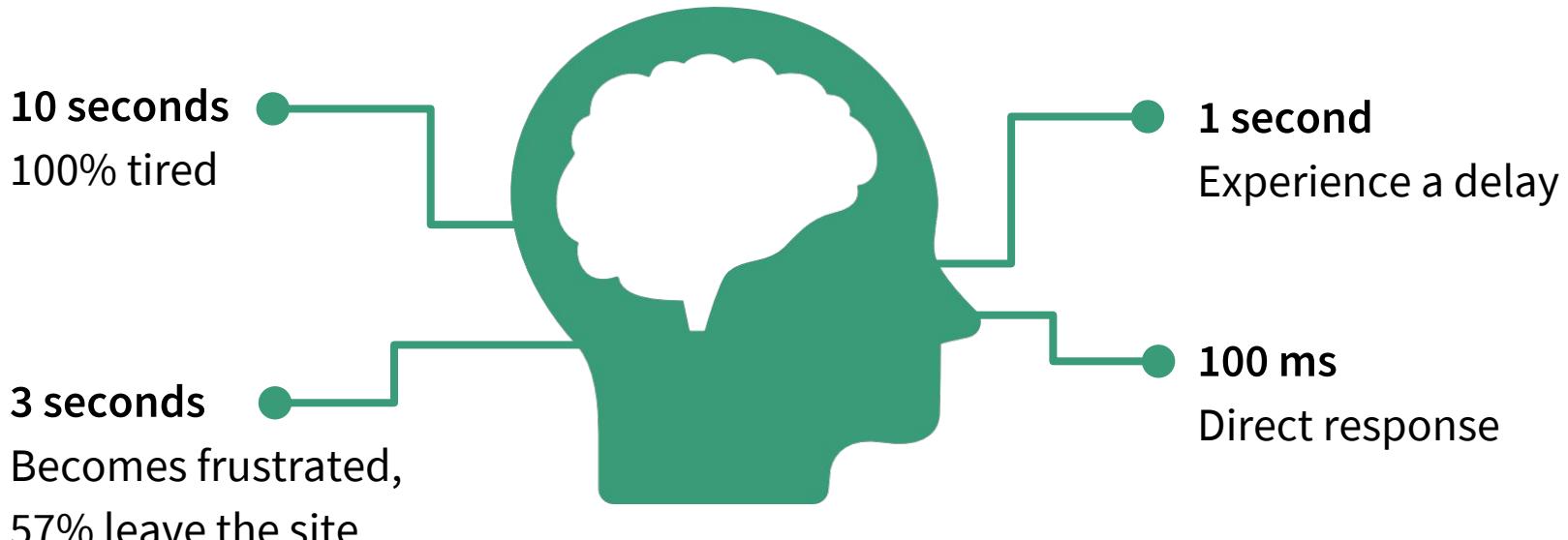




© Speedment, Inc.



Why are Delays a Problem?



Experience the Difference

10 s



3 s



1 s



0.1 s



Why are Web Delays a Problem?

Google lost 20% of traffic
with half a second delay

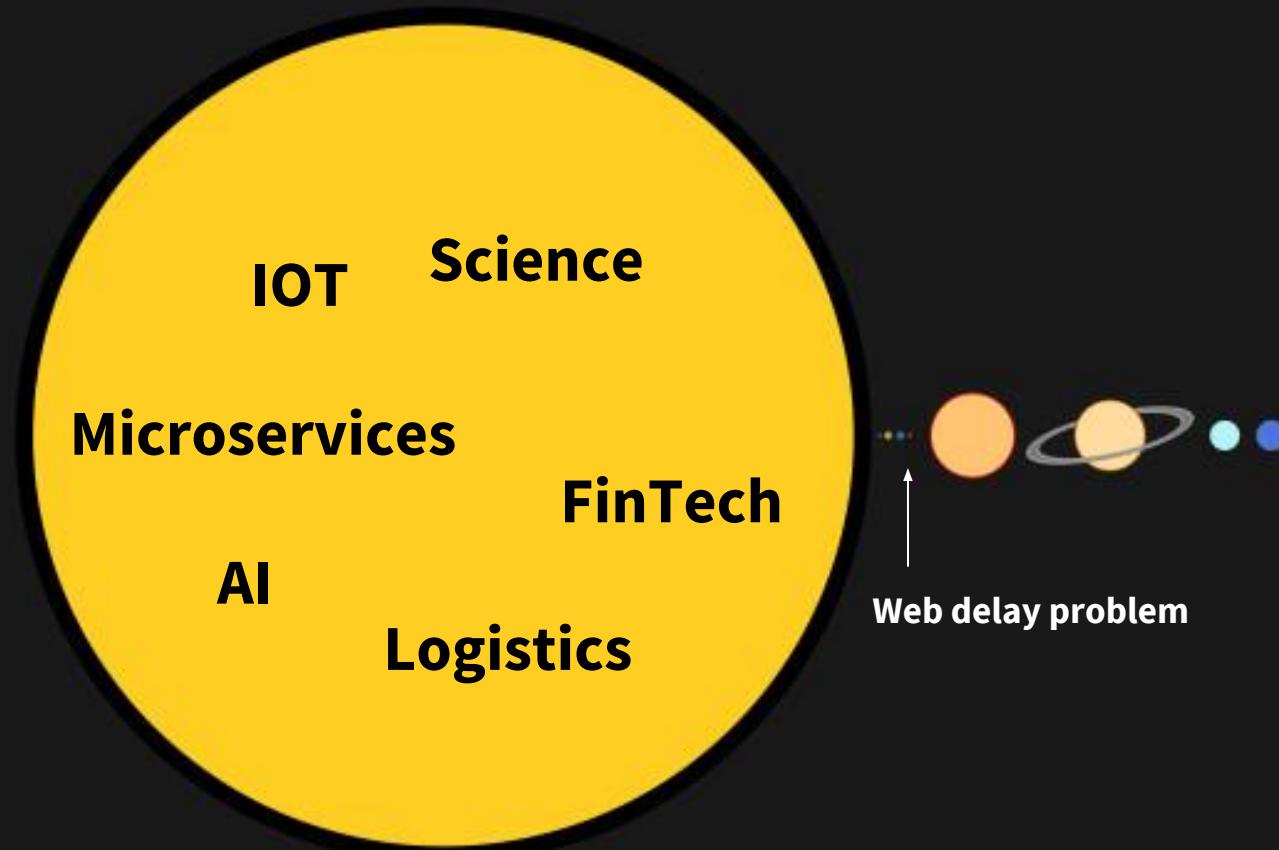
Less Page Views

Amazon lost 1% of sales for
every 100 ms delay

Less Revenue

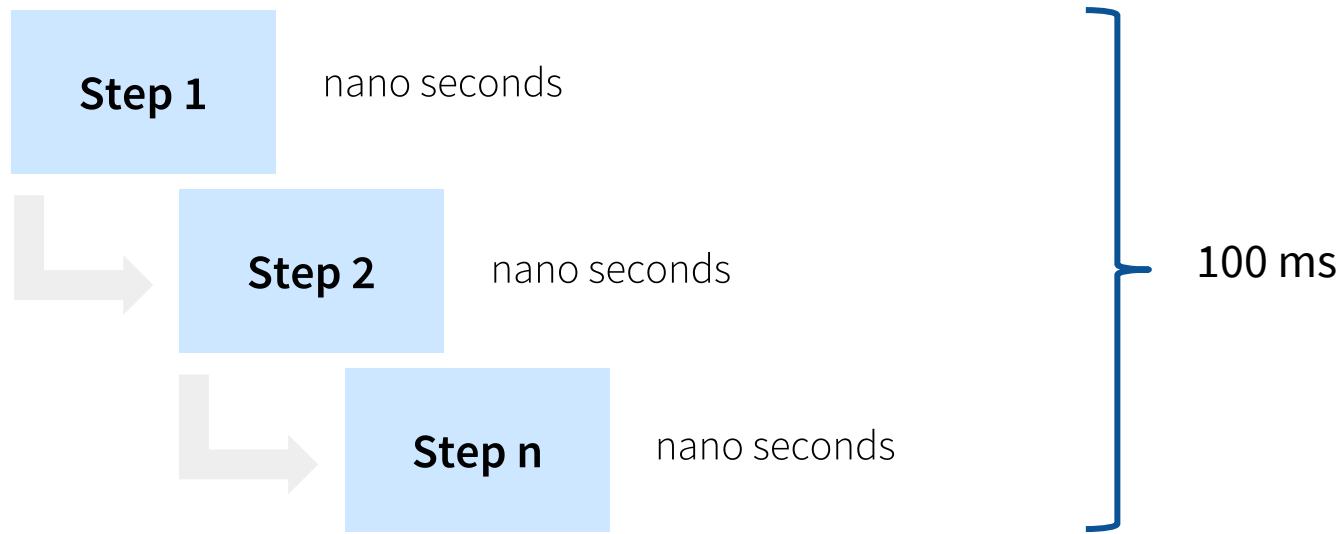
44% worry when paying
transactions take too long

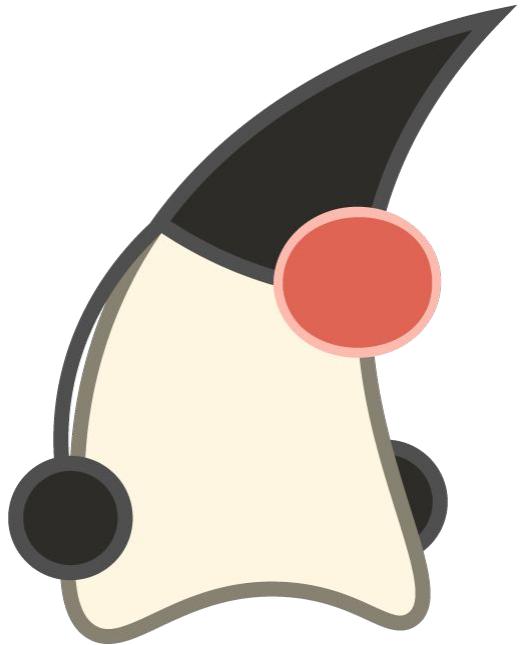
Destroys the brand



$\sum_{i=1}^n x_i = \sum_{i=1}^n y_i$ $X^3 = \begin{cases} x_2 + 2 \\ y_2 + 2x \end{cases}$ $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} X = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \Rightarrow \log(4 \cdot x - 7c) = u$ $P_A = P - C \cdot (x_2)^{1/n}$
 $\sum_{i=1}^n N_{i,i} + B_2 + f(P_1) - 2b_2$ $T_2 = t_2 - \Delta$ P_1, P_2, P_3 $\frac{\sin(p)}{4_2 + 2b + \sqrt{2}c}$ $d_{ij} = \cos(\theta_{ij}) - 2 \left[\frac{2}{\pi} - \frac{2}{3} \frac{\cos c}{\sqrt{2}} \right] \cos$ $x_2 + 4.92 = c$
 $x = Y \cdot \cos(\alpha) + 2$ $\frac{\pi}{2} - \frac{1}{2} - \frac{2b - \sqrt{x^2}}{2b + \cos(\theta)}$ $P_1 = 563L - x \sum_{i=1}^n 2b - \sqrt{x^2 - 2b^2}$ $b_{12} = \frac{C_{22} + e_{24}}{m_{12} + n_{12}}$ $c_1 = 22 (e=2)$
 $\boxed{10 \quad 2.943 - y}$ $\boxed{X = 4.91}$ $\boxed{Y = 2.81}$ $n = 5.43$ $\begin{cases} \int x_2 - y_{2n} = \frac{\pi}{3.98} \\ F \end{cases}$ $n = \begin{cases} (433.12 + 8332.12 = x^2 + 8.1433) \\ (x+y+abc = 5.631(83.4)) \end{cases}$ $22ci$ $\frac{29878}{-810} = \frac{010001}{01001} \quad \frac{01001}{01001}$
 $= m_{12}$ $n = 4.314$ $43c - 78.94n$ $\begin{cases} \text{int}(\cos(x) + \sin(y)) \\ \tan(\pi \cdot b) \end{cases}$ $B?$ Ah $\begin{cases} px+2 \\ B_3 \end{cases}$ $D(x) = 2 + 3 + 4.31447$
 $\boxed{1220}$ $\boxed{1001}$ $\boxed{1001}$ $\partial + b + c = 4x^2$ $\begin{cases} h-a \\ \sqrt{2454.56} = 2x + 2e^{-n^{10}} \end{cases}$ $L = \frac{1}{2}$
 $\sum_{i=1}^n \log x_{10} = \ln(n) - 2a$ $n = \sqrt{3x + 4y}$ $\sum_{i=1}^n i = \cos(x) \cdot ab^2$ $\begin{cases} A \\ 0 \end{cases} \quad \sqrt{a^2 + b^2} = x^2$ $\begin{cases} 22 \\ ux \end{cases}$ $\begin{cases} \square + \square = 5.43294 \\ V=22 \quad (a+b) \end{cases}$
 $- (c_p + b)$ $\sin(\beta) = N_1$ 222.856000 $\pi = 3.14$ $\begin{cases} x^2 + y^2 = ab + 4c \\ x^2 + y^2 = ab + 4c \end{cases}$ $c(x, y) \begin{cases} xy = c \\ cx - cy = 25^2 \\ \pi = c \end{cases}$ $\begin{cases} AB \\ X \end{cases}$
 $76+x$ $\sin(B) = N_2$ $\sum \begin{cases} x^2 + y^2 = 54 \\ 2x + b = -xy \\ 432x = y + ab \end{cases}$ $\beta = 90$ $A=B$ $\begin{cases} 24+x \\ y \end{cases} + \frac{a^2 + b^2}{c} + \frac{2}{x} \rightarrow g$
 $\vdots = NM$ $\tan(\beta \cdot \beta) = F$ $\begin{cases} A \\ B \end{cases}$ $\begin{cases} 24+x \\ y \end{cases} + \frac{a^2 + b^2}{c} + \frac{2}{x} \rightarrow g$ $m_{12} = 384 + n^{10}v$ $(x^2 + 34x + c^2)$
 $\begin{cases} N_1, M_2 \\ N_2, N_3 \end{cases}$ $\begin{cases} j \\ M \end{cases}$ 6 688 $\begin{cases} l = 5c + c ? \\ l = (b^2 + 4c) \cdot \sqrt{c} ? \end{cases}$ $\begin{cases} x \leq 549 \\ e \end{cases}$
 $u \overline{1} (2+e2) \cos(x) + y + 4.5653288 = MN + k_2$ $\begin{cases} l = 5c + c ? \\ l = (b^2 + 4c) \cdot \sqrt{c} ? \end{cases}$ $x = 9.20$ $\begin{cases} u=14! \\ 0 \end{cases} \quad \begin{cases} \sum N_{30} \cdot x - \frac{1}{2} \int [384 + x^2 + pab] \end{cases}$
 $2.943 \quad 2x^2 + 4.51y + 7.62z = MN$ $P(N) = f_{xy} (4.5653281 + x_2 + y_2 = \sqrt{MN})$ $\sqrt{42+2c} = nm_2 \lim(x)$
 $\begin{cases} u=4 \\ v=4 \end{cases}$ $P(u) = \sqrt{h^2 + (1 - \frac{x^2}{z})^2}$ $\lim(x) = \log m - 1(x)$ $\begin{cases} 010112 \\ 010002 \end{cases} \quad \begin{cases} \beta = 9 + x^2 + y^2 \\ 810 \end{cases}$

Latency Requirement Break-down



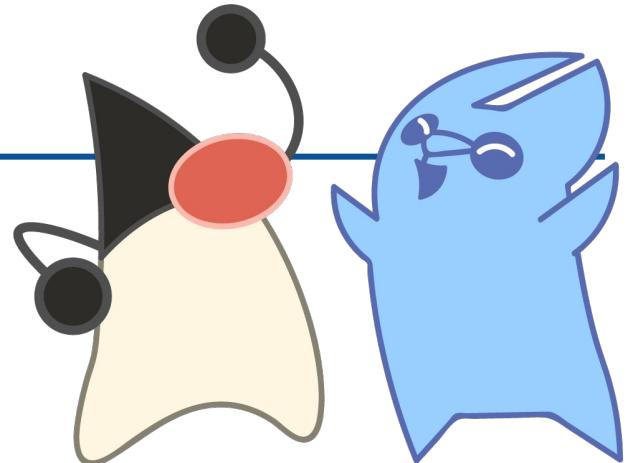
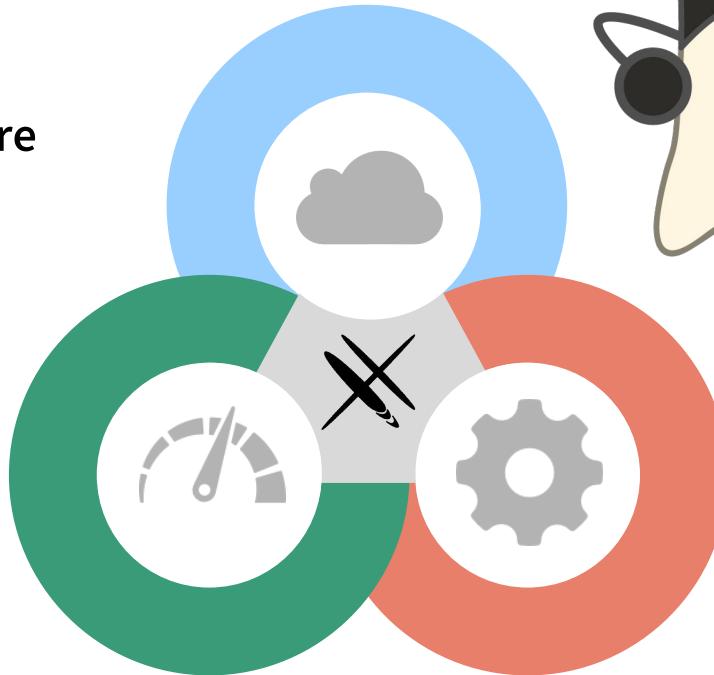


Speedment Solution

Use Existing Infrastructure

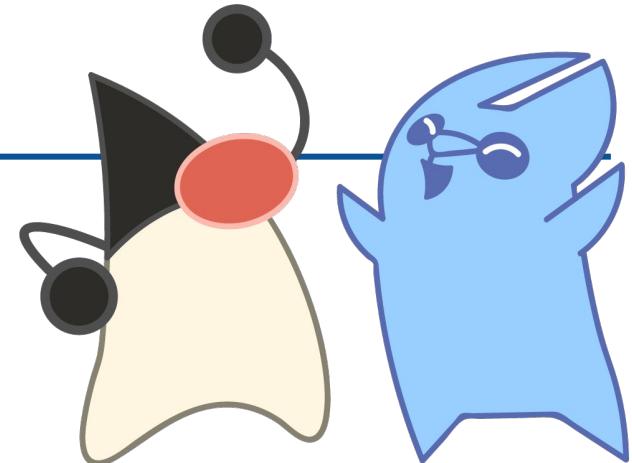
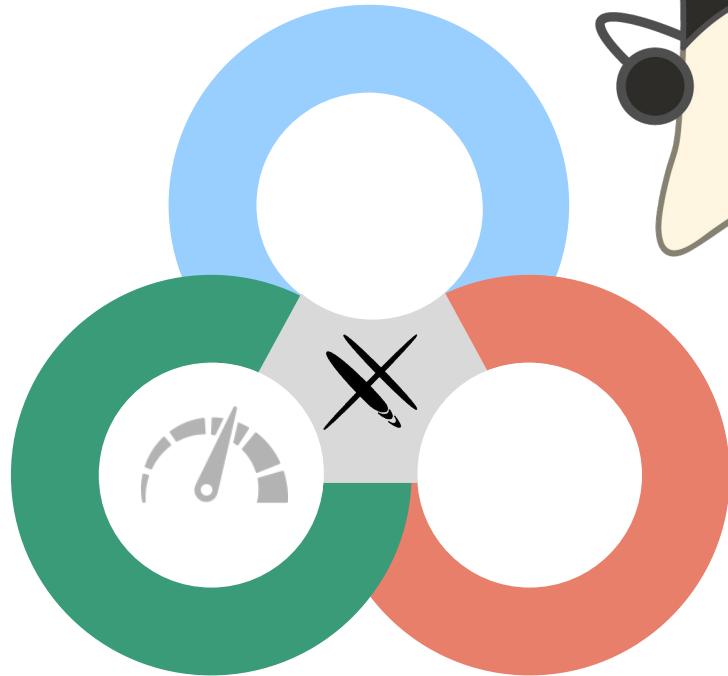
Ultra-low Latency

Ease of Development

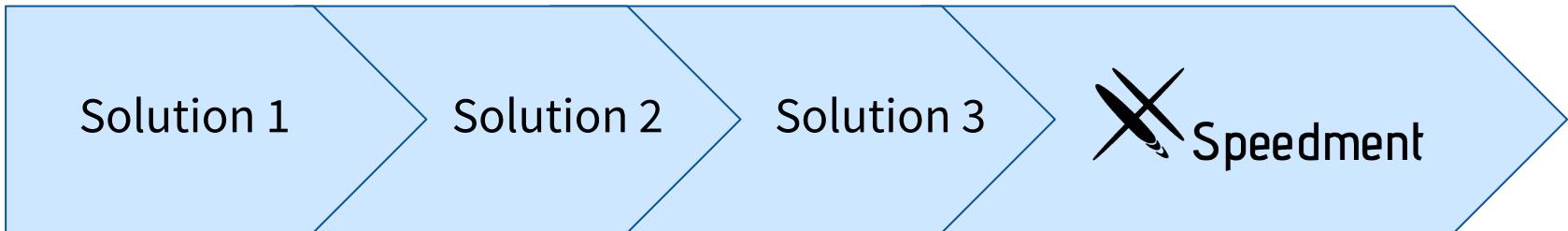


Speedment Solution

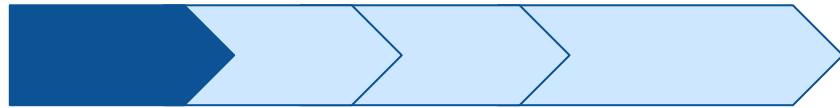
Ultra-low
Latency



How to Gain Ultra-Low Latency ?

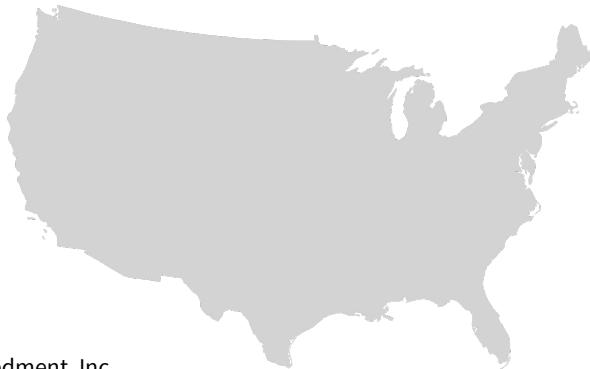


Cluster of Nodes



San Francisco to New York speed of light latency

> 45 ms for fiber



TCP round trip latency with two Linux hosts connected directly with 10Gb/s Ethernet

- Some tweaks 40 μ s
- Busy polling and CPU affinity 30 μ s
- Expert mode ~25 μ s

>25 μ s

+ Delays from routers, switches, cloud solutions

>100 μ s

Different Processes



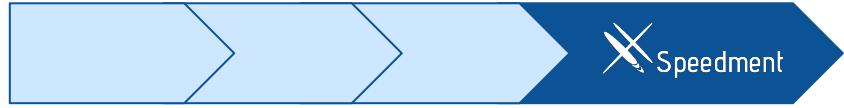
- On the same machine
- Inter-Process Communication is in the milliseconds
- Context Switch -> L1, L2, L3 + TLB affected

In-Memory Object



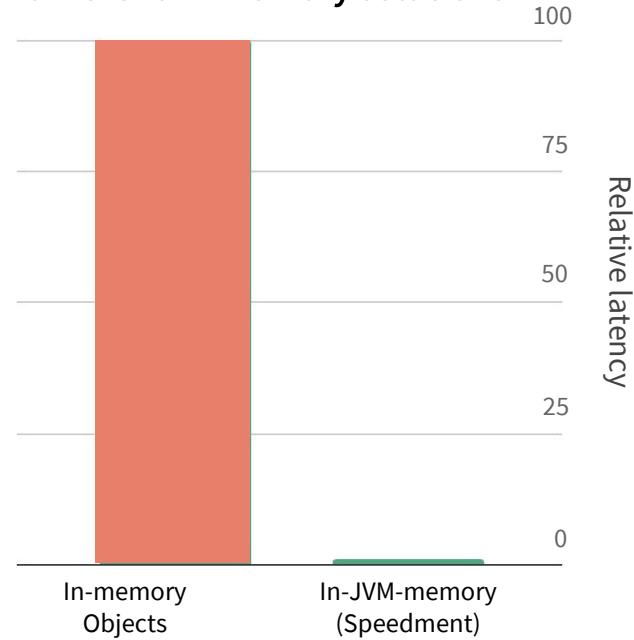
- Garbage Collect
- Latency > 1 s
- GC -> L1, L2, L3 + TLB totally ruined

In-JVM-Memory

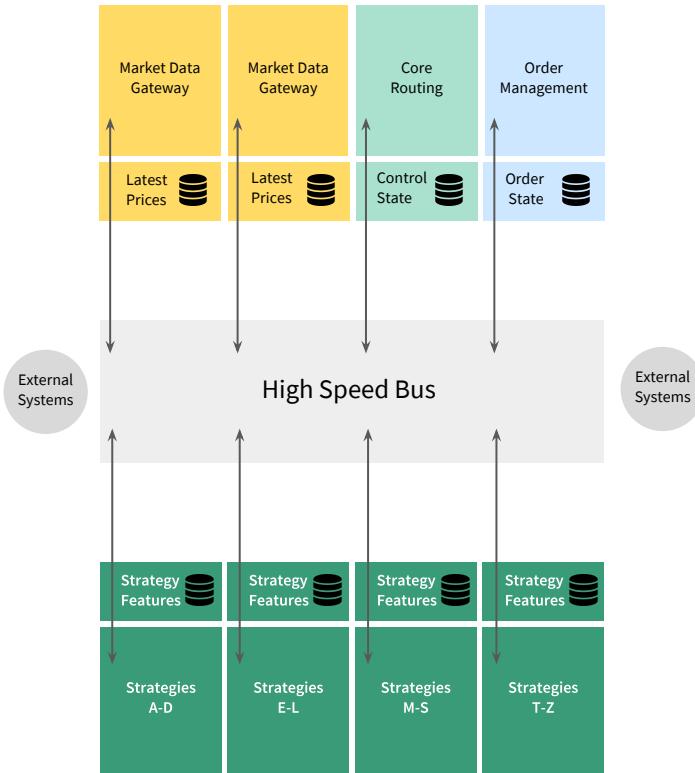


- Main Memory Read ~100 ns
- Volatile read
- L3 ~20 ns
- L2 ~7 ns
- L1 ~0.5 ns
- CPU Registers

Latency comparison between different in-memory solutions

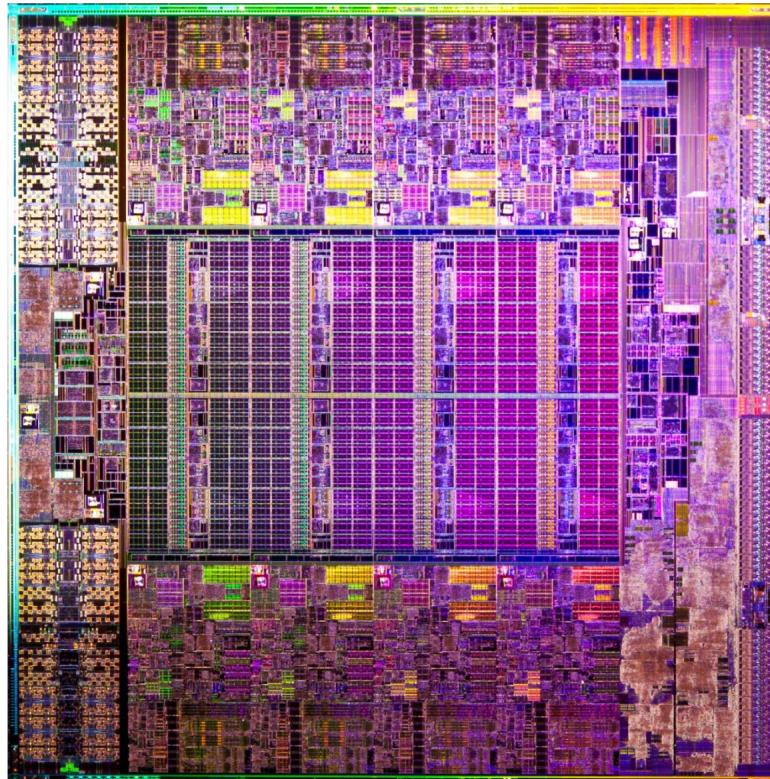


What Makes Speedment Speedy?



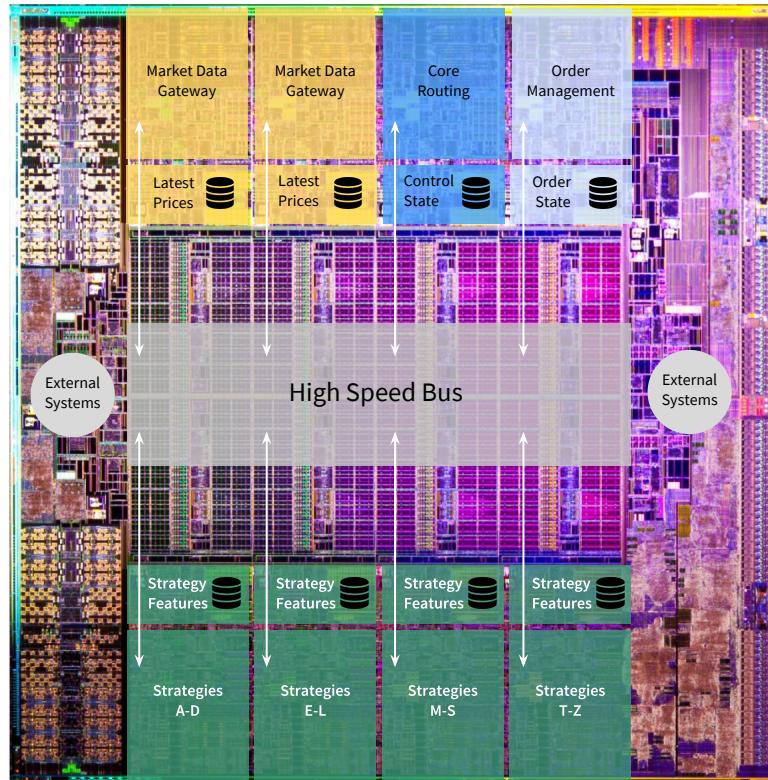
What Makes Speedment Speedy?

Multi Core
Intel CPU

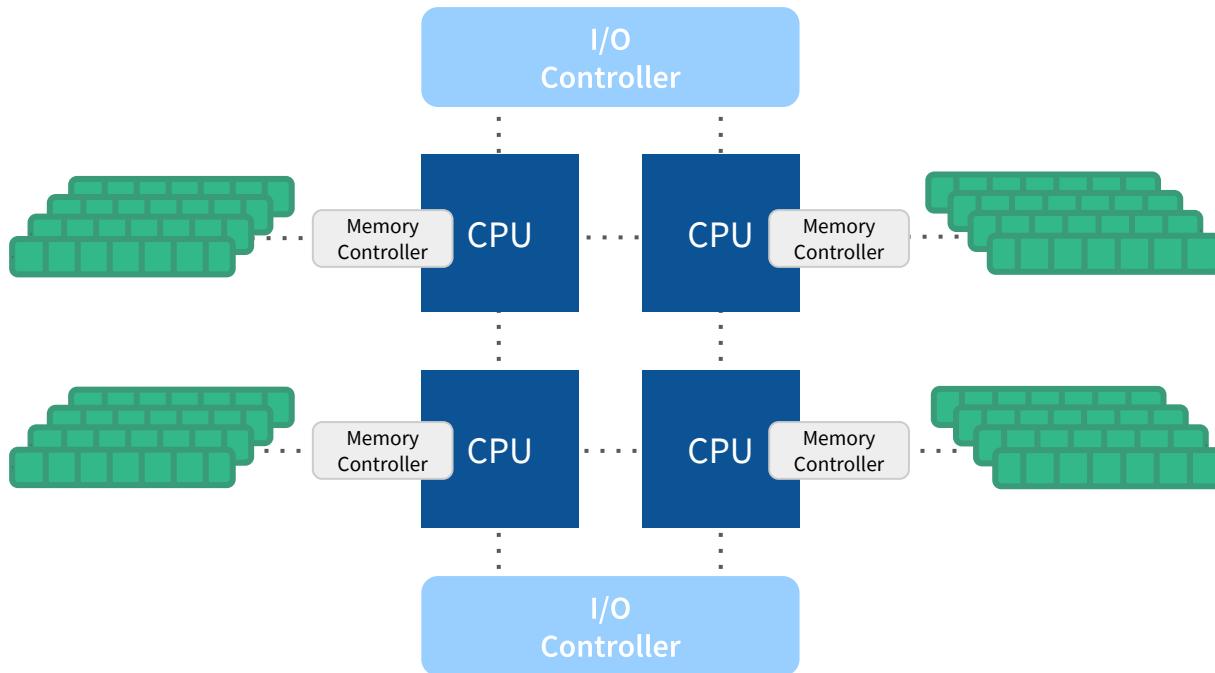


What Makes Speedment Speedy?

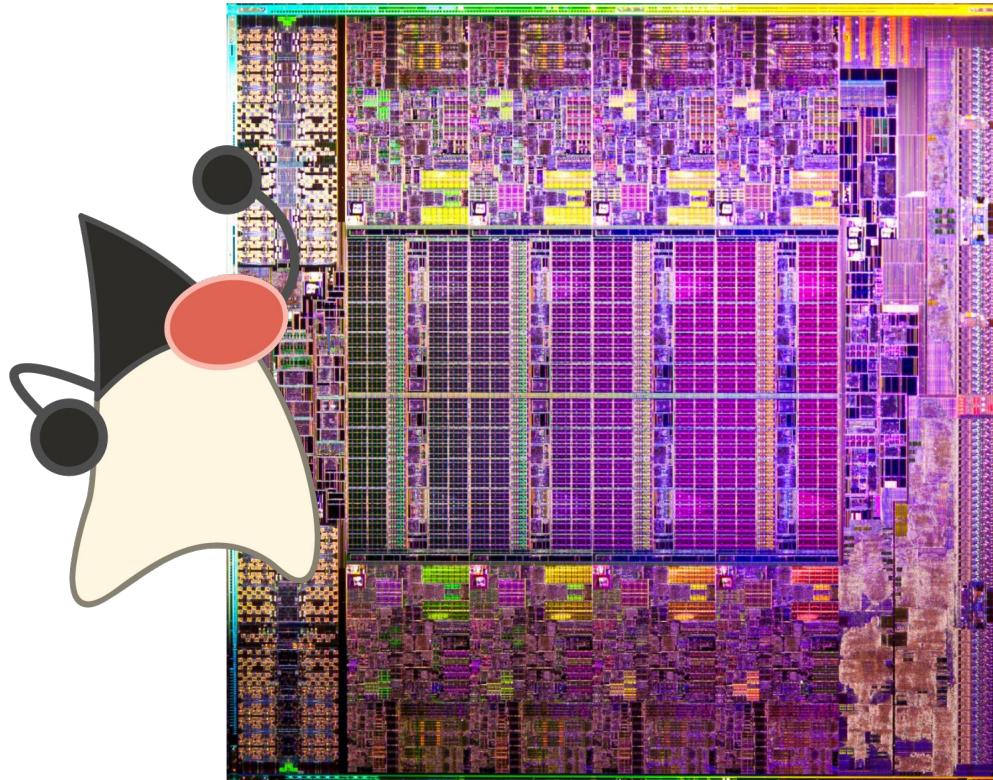
Multi Core
Intel CPU



What Makes Speedment Speedy?



Conclusion: In-JVM-Memory



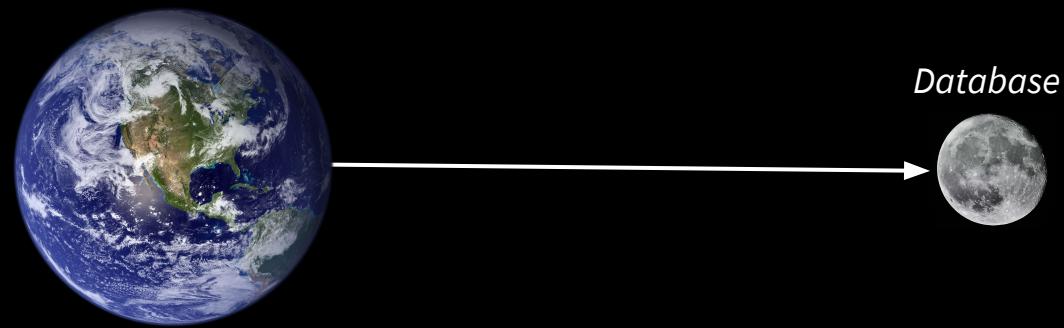
The Difference

Reaching L1 CPU cache takes **0.5 ns**. Corresponding to the time it takes for light to travel ~4 inches.

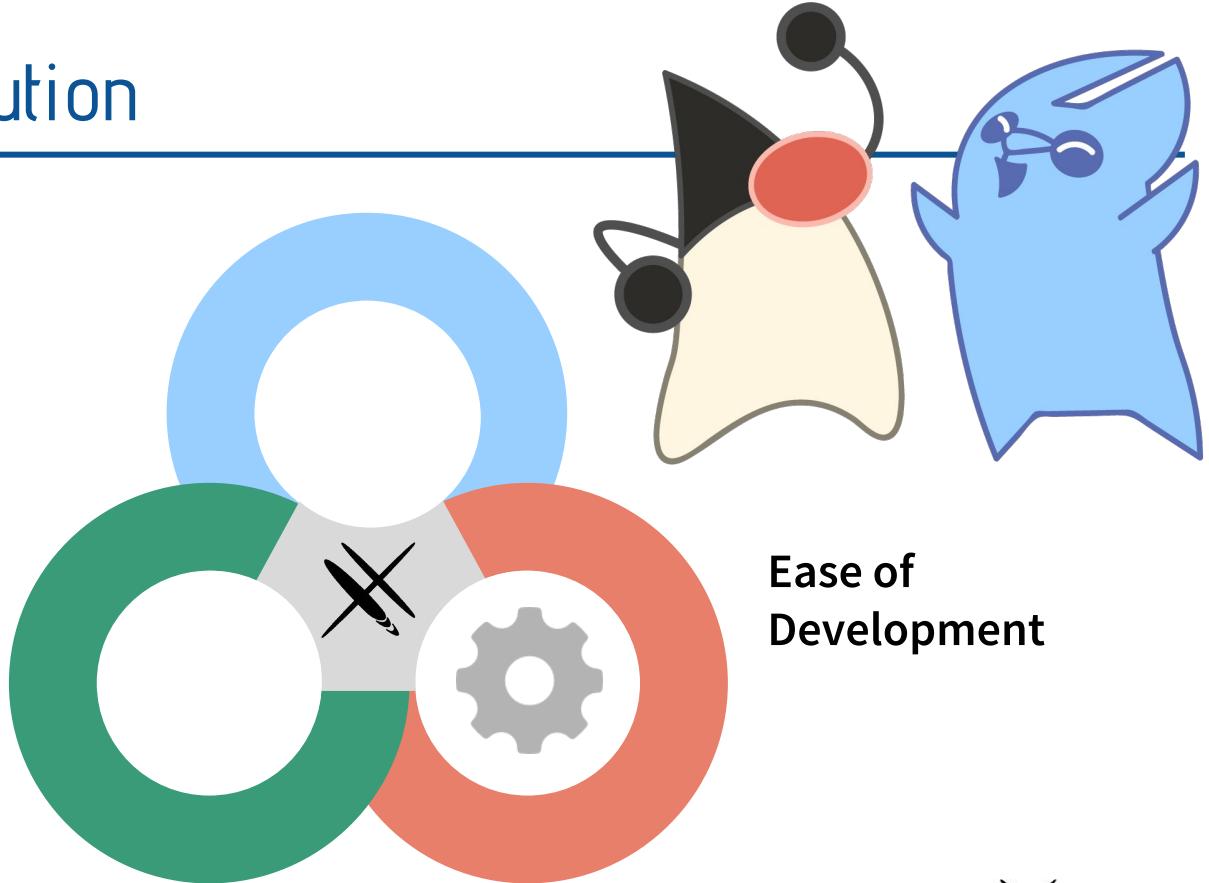
During the time a database makes a **one second query**,
how far will the light move?

CONCLUSION

Don't place your data on the moon - **keep it close by using In-JVM-Memory technology!**

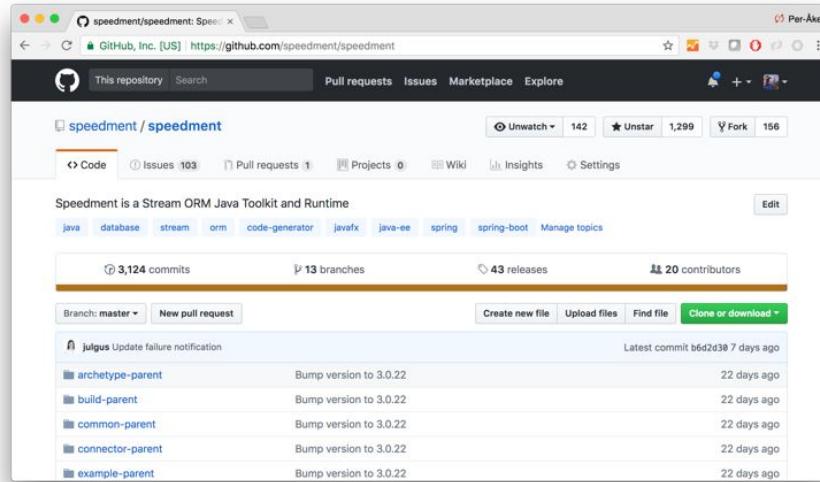


Speedment Solution



Ease of Development

Use Existing API
Standard Java Stream



Oracle Java Magazine Article



//databases /



PER MINBORG

Database Actions Using Java 8 Stream Syntax Instead of SQL

Speedment 3.0 enables Java developers to stay in Java when writing database applications.

Why should you need to use SQL when the same semantics can be derived directly from Java 8 streams? If you take a closer look at this objective, it turns out there is a remarkable resemblance between the verbs of Java 8 streams and SQL commands, as summarized in [Table 1](#).

Streams and SQL queries have similar syntax in part because both are declarative constructs, meaning they describe a result rather than state instructions on how to compute the result. Just as a SQL query describes a result set rather than the operations needed to compute the result, a Java stream describes the result of a sequence of abstract functions without dictating the properties of the actual computation.

The open source project Speedment capitalizes on this similarity to enable you to perform database actions using Java 8 stream syntax instead of SQL. It is available on GitHub under the business-friendly Apache 2 license for open source databases. (A license fee is required for commercial databases.) Feel free to clone the entire project.

About Speedment

Speedment allows you to write pure Java code for entire database applications. It uses lazy evaluation of streams, meaning that only a minimum set of data is actually pulled from the database into your application and only as the elements are needed.

In the following example, the objective is to print out all `film` entities having a rating of PG-13 (meaning "parents are strongly cautioned" in the US). The films are located in a database table represented by a Speedment Manager table.

SQl COMMAND	JAVA 8 STREAM OPERATIONS
FROM	stream()
SELECT	map()
WHERE	filter() (BEFORE COLLECTING)
HAVING	filter() (AFTER COLLECTING)
JOIN	flatMap() OR map()
DISTINCT	distinct()
UNION	concat(s0, s1).distinct()
ORDER BY	sorted()
OFFSET	skip()
LIMIT	limit()
GROUP BY	collect(groupingBy())
COUNT	count()

Table 1. SQL commands and their counterpart verbs in Java 8 streams



34

40

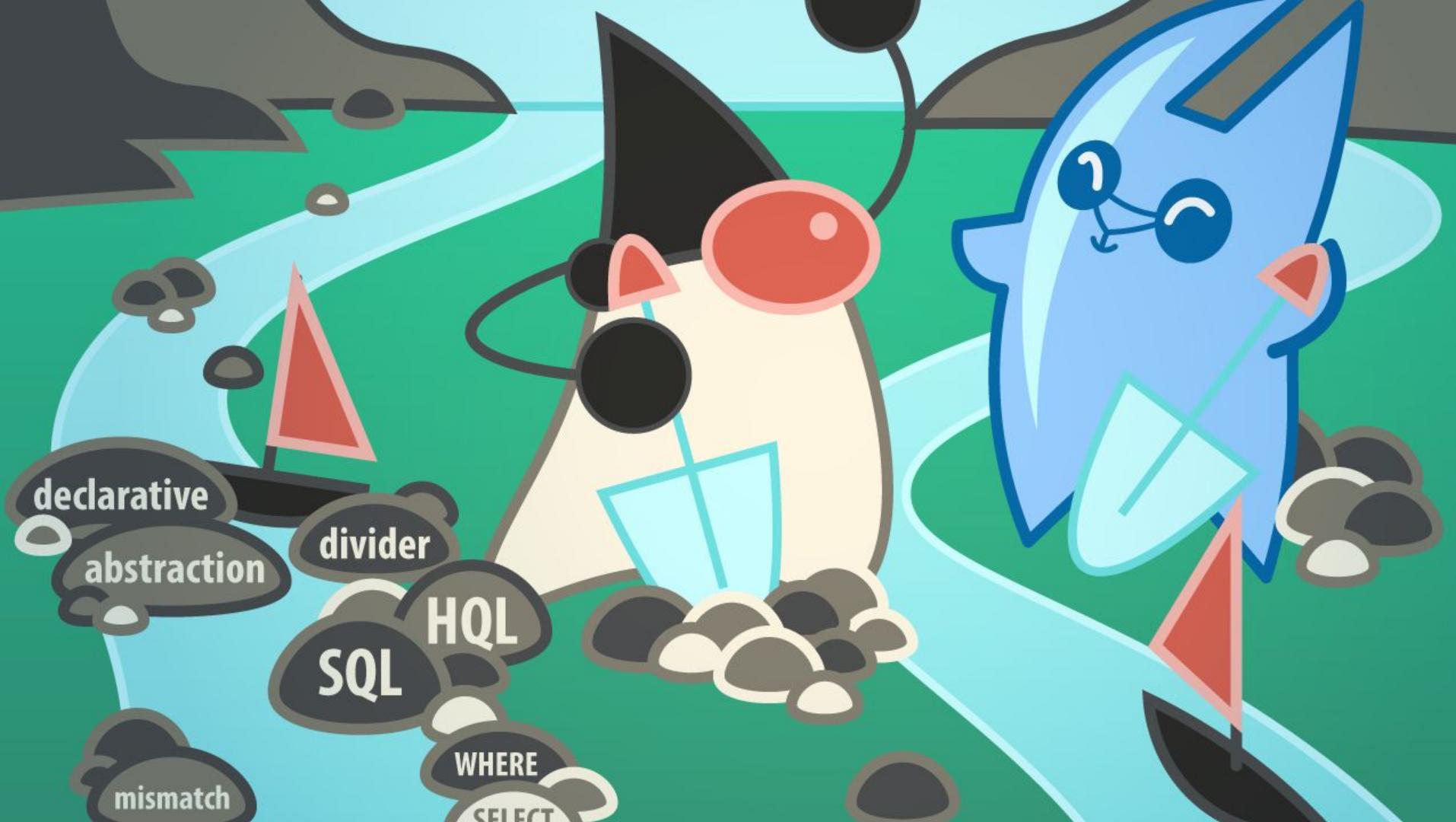
Comparison Between SQL and Stream Operations

SQL	Java Stream Operations
FROM	stream()
COUNT	count()
LIMIT	limit()
DISTINCT	distinct()
SELECT	map()
WHERE	filter() <i>before collecting</i>
HAVING	filter() <i>after collecting</i>
JOIN	flatmap() or map()
UNION	concat(s0, s1).distinct()
ORDER BY	sorted()
OFFSET	skip()
GROUP BY	collect(groupingBy())

Declarative Constructs in Both SQL and Stream

```
SELECT * FROM FILM  
WHERE RATING = 'PG-13'
```

```
films.stream()  
.filter(Film.RATING.equal(Rating.PG13))
```



declarative

abstraction

divider

HQL

SQL

mismatch

WHERE

SELECT

Java Streams Make Execution Even Faster

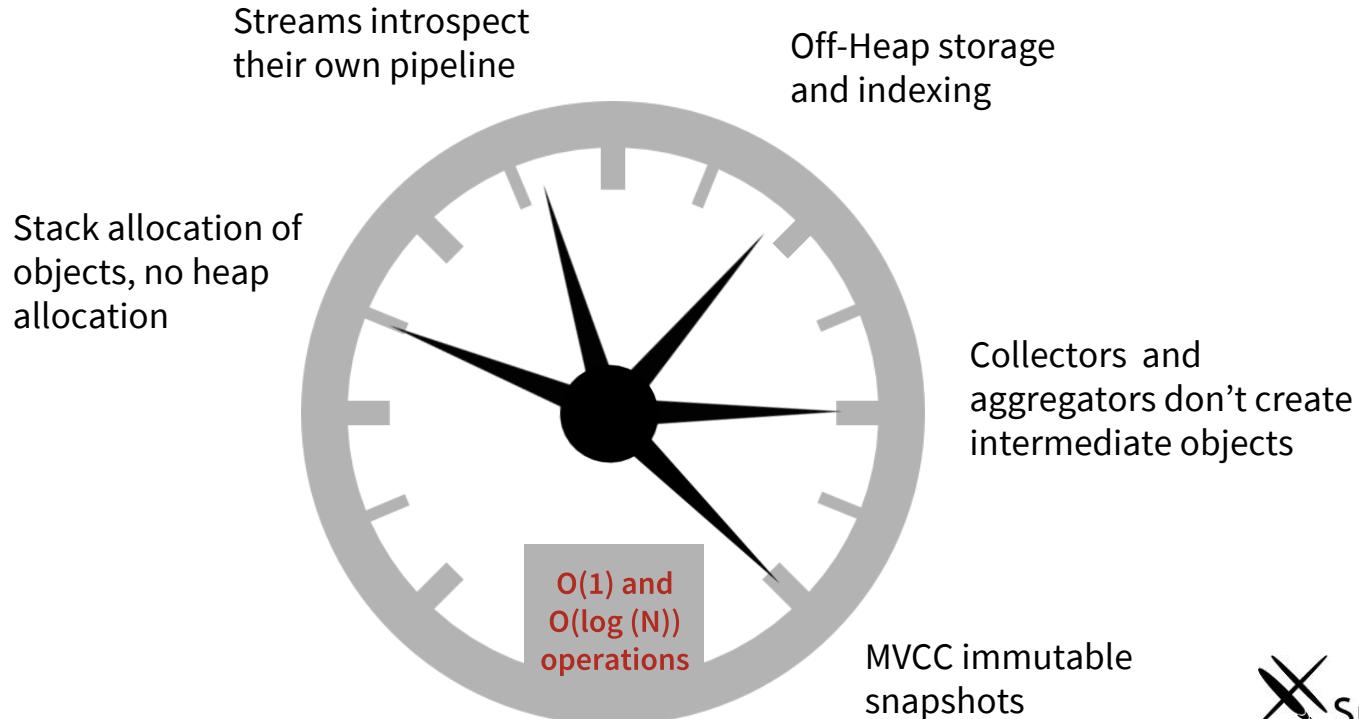


Java Streams Make Execution Even Faster

```
films.stream()  
    .filter(Film.RATING.equal(Rating.PG13))  
    .collect(toJsonLengthAndTitle());
```

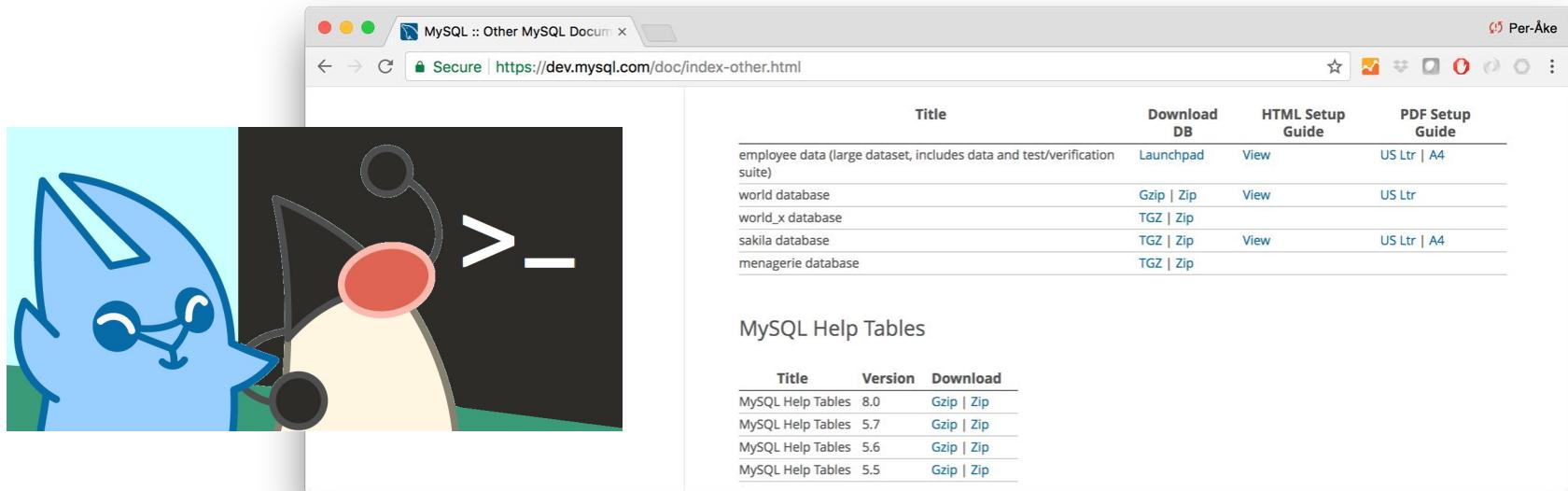
index	film_id	length	rating	year	language	title
[0]	0	267	267	0	0	0
[1]	267	0	0	267	267	267
[2]	523	523	523	523	523	523
address		film_id	length	rating	year	language
[0]	0	1	4	12	16	20
[267]	2	69	PG-13	2006	1	ACAD...
[523]	3	134	PG-13	2006	1	ADAP...

... And Faster



Speed Demo

Sakila – Open Source Film Database for MySQL



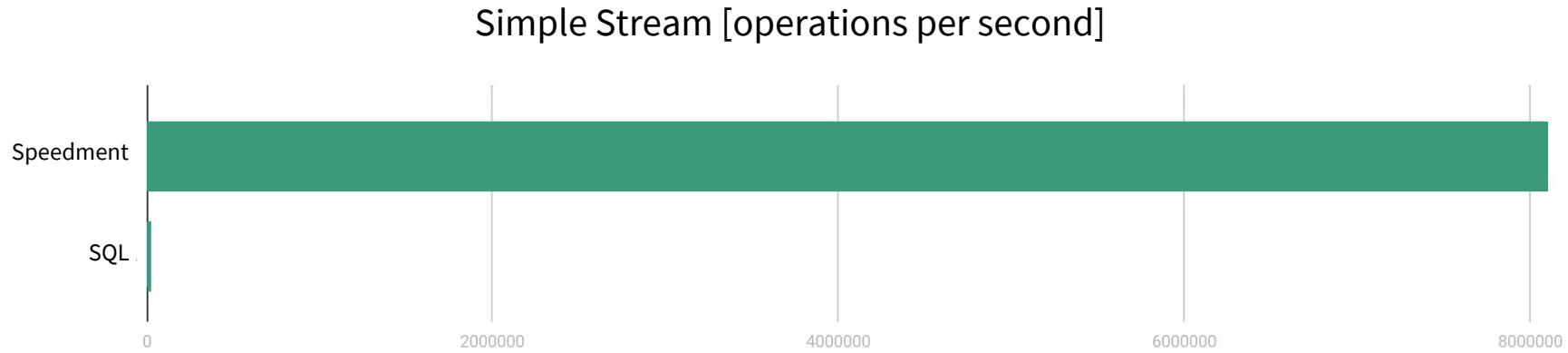
A screenshot of a web browser displaying the MySQL documentation for "Other MySQL Datasets". The URL is <https://dev.mysql.com/doc/index-other.html>. The page features a large, semi-transparent watermark of a cartoon character (a blue cat-like creature) on the left side. The main content includes a table of datasets and their download links, and a section for "MySQL Help Tables".

Title	Download DB	HTML Setup Guide	PDF Setup Guide
employee data (large dataset, includes data and test/verification suite)	Launchpad	View	US Ltr A4
world database	Gzip Zip	View	US Ltr
world_x database	TGZ Zip		
sakila database	TGZ Zip	View	US Ltr A4
menagerie database	TGZ Zip		

MySQL Help Tables

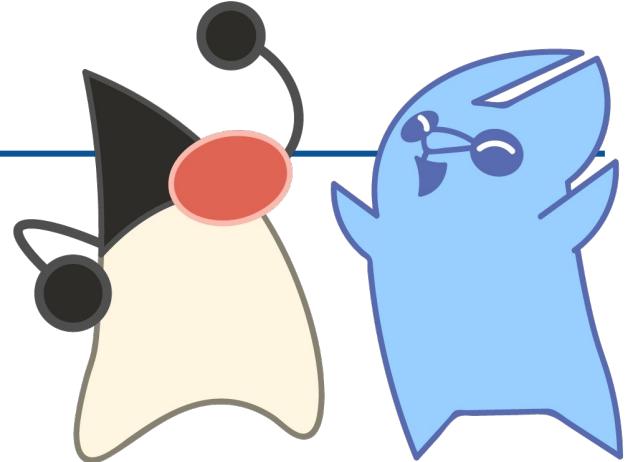
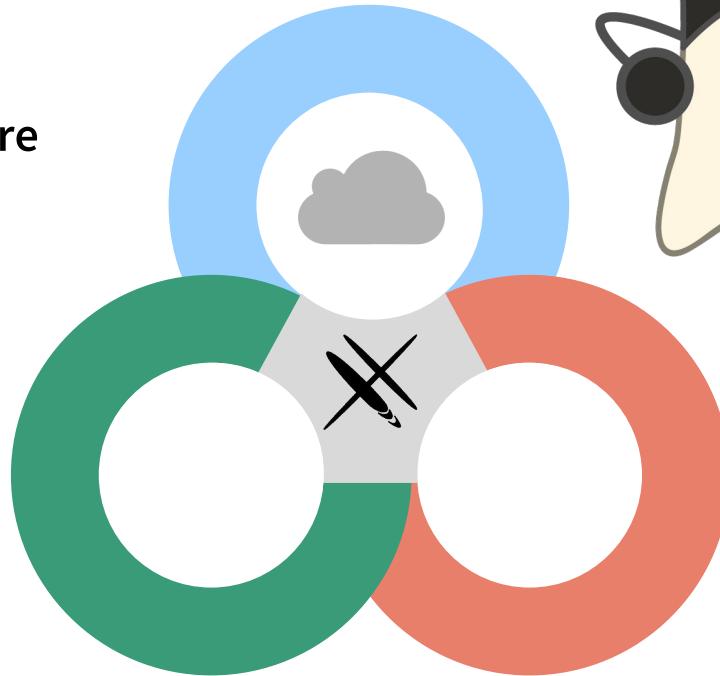
Title	Version	Download
MySQL Help Tables	8.0	Gzip Zip
MySQL Help Tables	5.7	Gzip Zip
MySQL Help Tables	5.6	Gzip Zip
MySQL Help Tables	5.5	Gzip Zip

The Difference



Speedment Solution

Use Existing
Infrastructure



Easy Integration - Datasource

ORACLE®
DATABASE



IBM® AS/400®



PostgreSQL



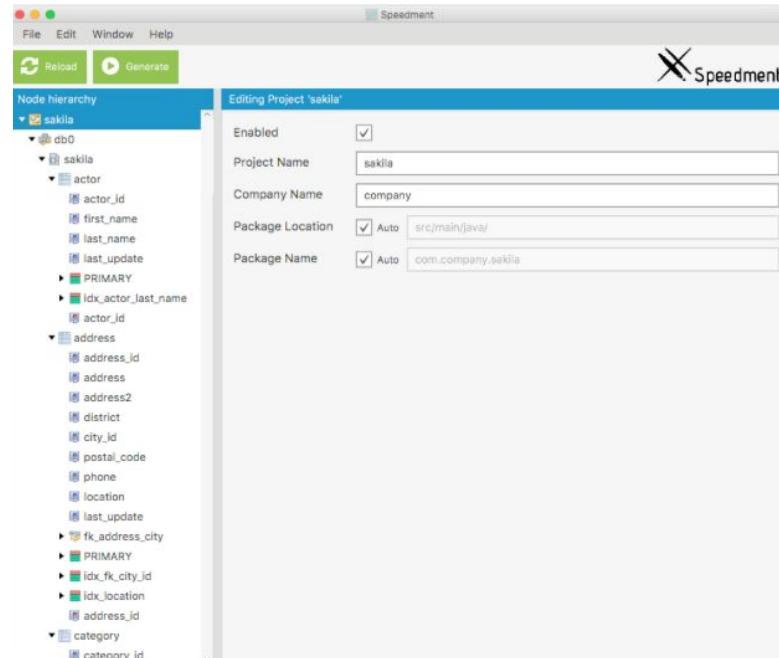
MySQL®



Google Sheets



Microsoft®
SQL Server



Deploy Anywhere



IBM **Bluemix**™



On Premise



docker



Google Cloud Platform



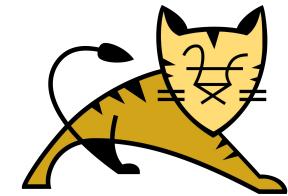
kubernetes

IDE Integration



NetBeans

Web Server Integration



Application API

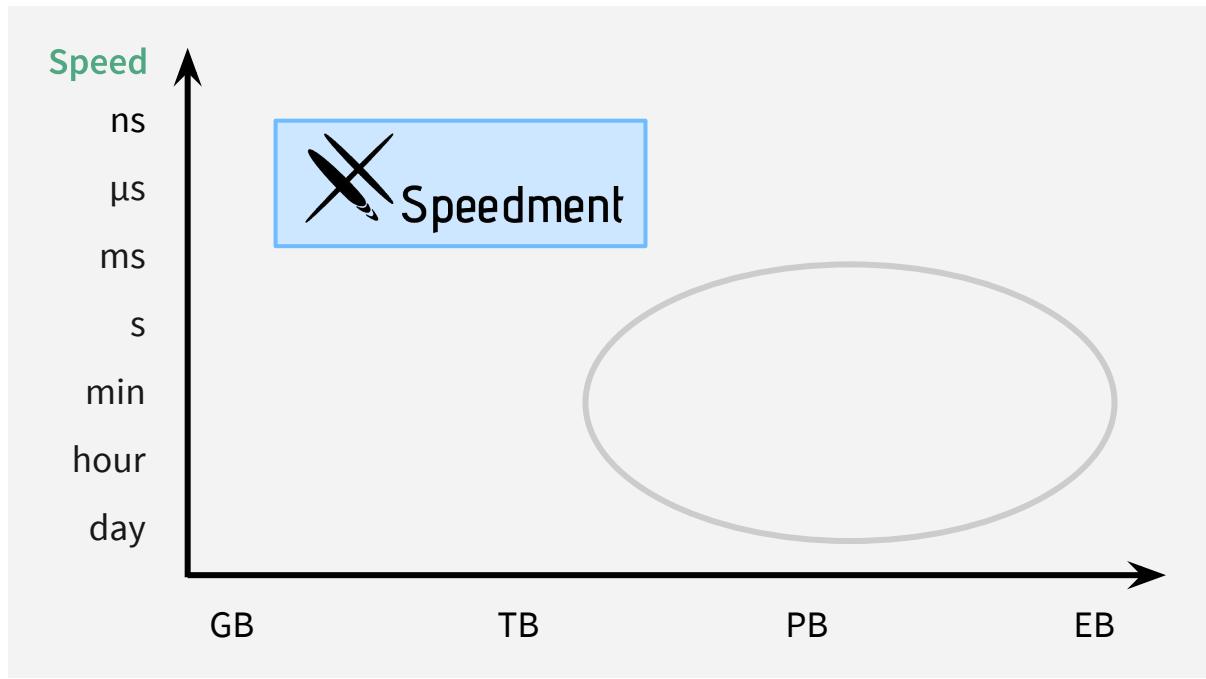
8 9



10

{ RESTful }

Position



Use Case: Transportation Company

PROBLEM

Demanding queries on large dataset are slow and heavy

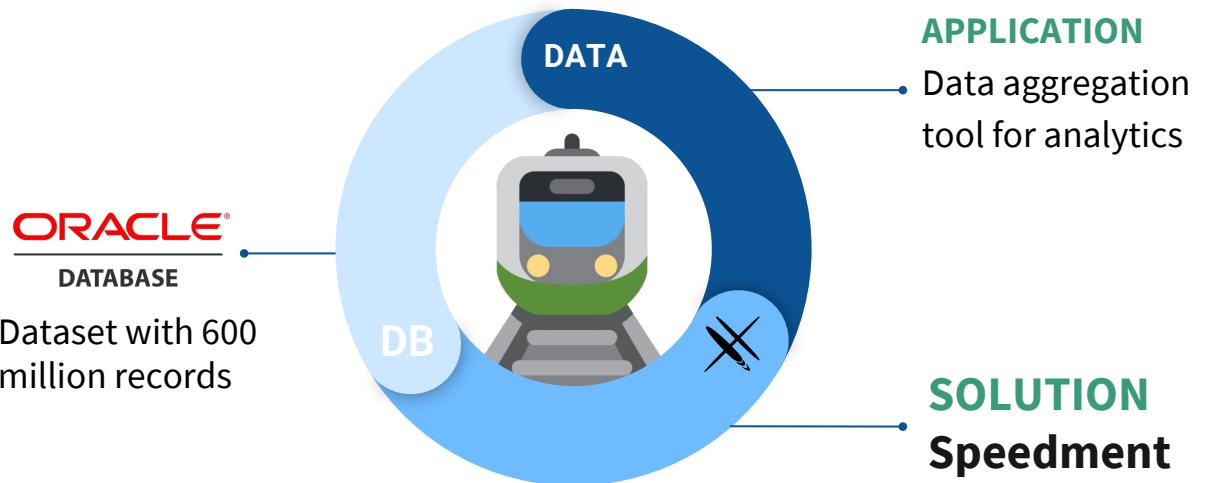
BEFORE

102 s
at high cost

AFTER

1 s
at much lower cost

**100x
faster with
Speedment**



Use Case: Telecom

PROBLEM

- Long delays and table locks
- 5 minutes to complete problematic queries

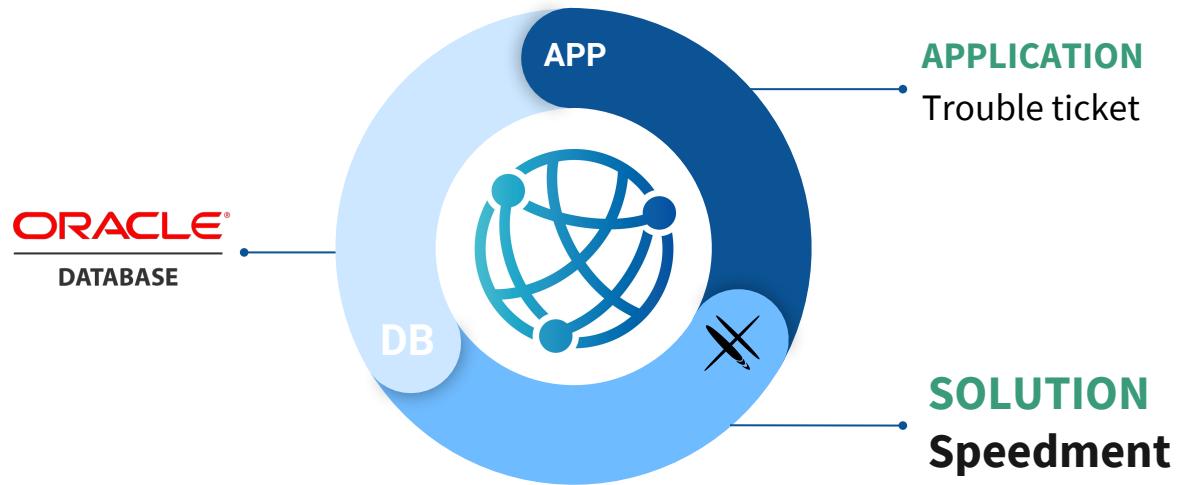
BEFORE

5 min

AFTER

1 s

**300x
faster with
Speedment**



Use Case: Spare parts

PROBLEM

Speed and cost reduction

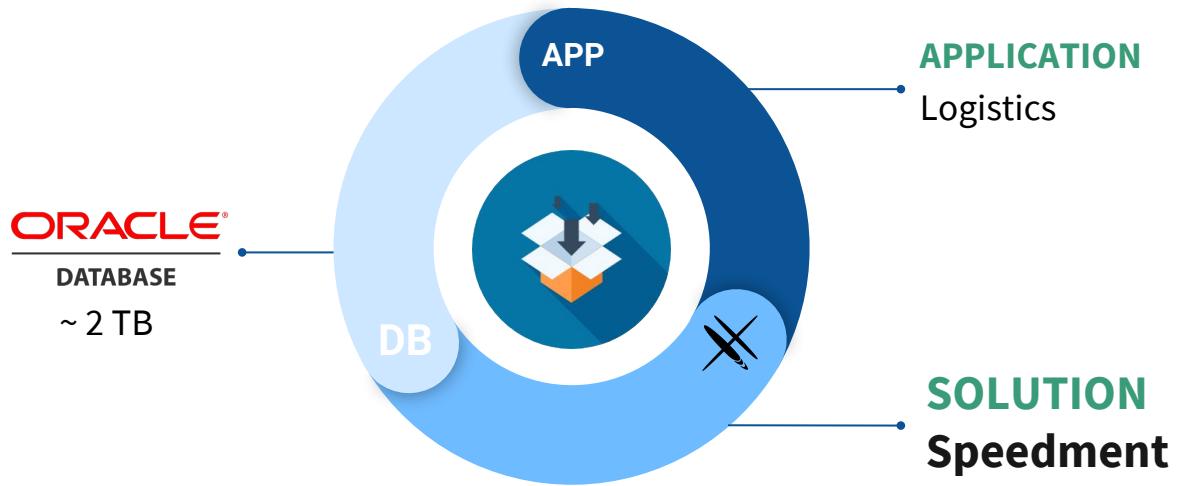
BEFORE

1 day
at high cost

AFTER

4 h
at lower cost

**6x
faster with
Speedment**



Use Case: Healthcare Industry

PROBLEM

- Analytics queries have unacceptable delays
- Unable to migrate to other DB technology

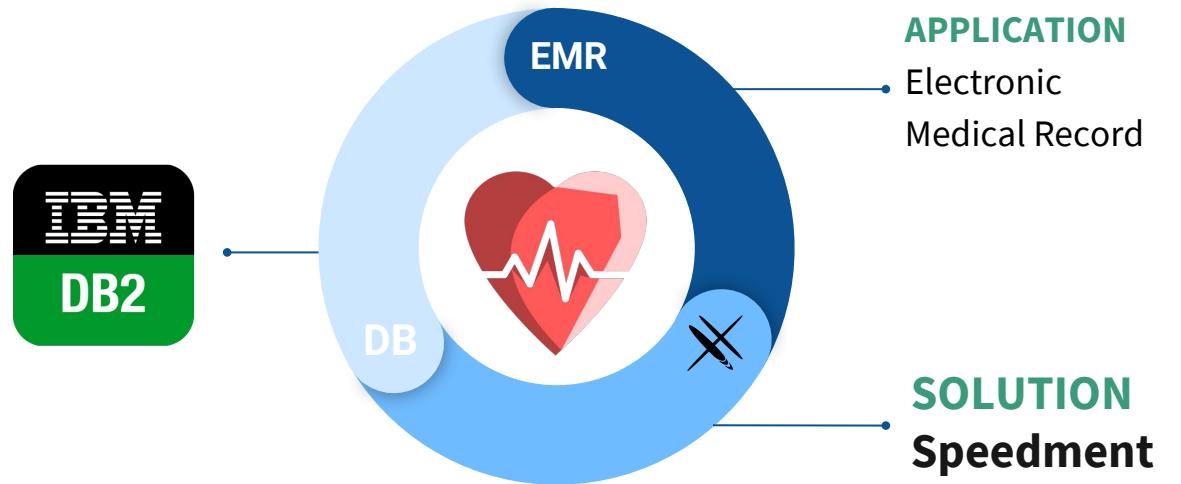
BEFORE

1 min

AFTER

1 s

**60x
faster with
Speedment**



Use Case: FinTech

PROBLEM

Interactive queries
are slow

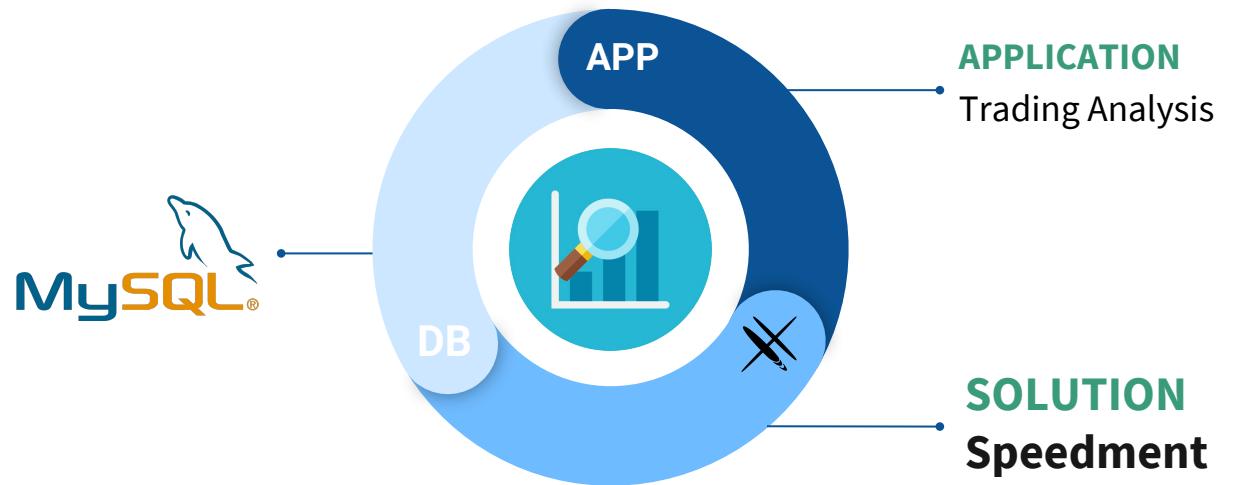
BEFORE

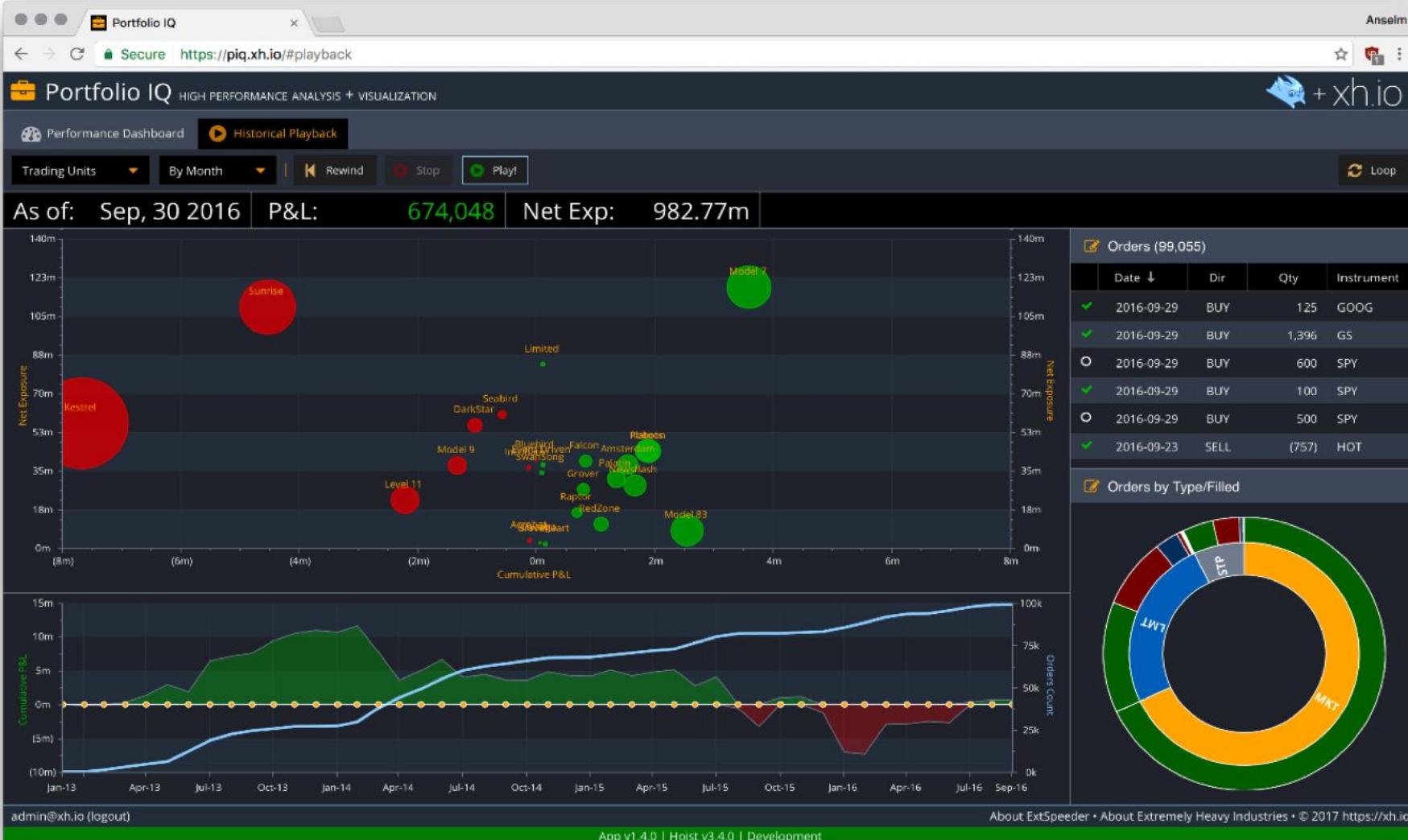
4 s

AFTER

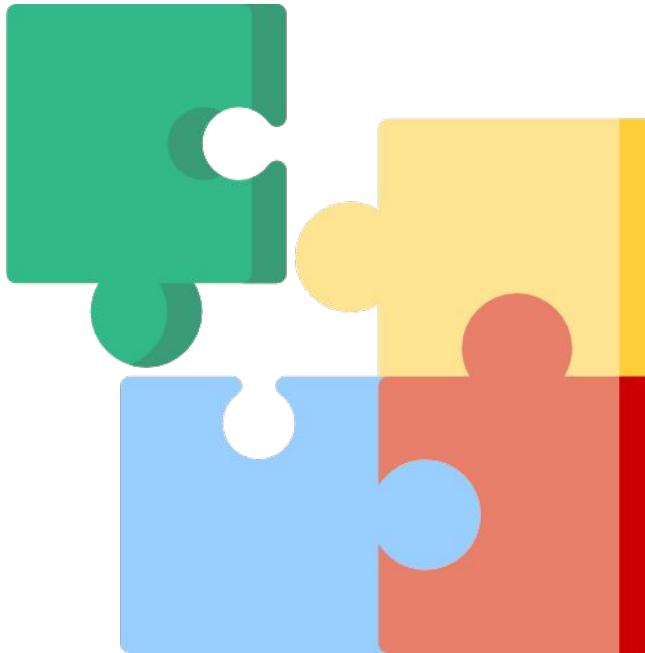
100 ms

40x
faster with
Speedment





Stepwise Introduction



The Initializer

The Initializer allows configuration of a new Speedment project in seconds.

To try Enterprise features such as enterprise databases and *In-Memory Acceleration*, a free 30-day trial is available.

The screenshot shows the Speedment Initializer interface. On the left, there's a sidebar with the title "The Initializer". The main area contains configuration fields for a database project:

- Database Type:** MySQL (selected), PostgreSQL, MariaDB, Oracle*, DB2*, AS400*, SQL Server*
- JDBC Driver Version:** 5.1.42
- Plugins:** Enums, Spring, JSON
- In-memory Acceleration:** Enable* (selected), Disable
- GroupId:** com.example
- ArtifactId:** demo
- Version:** 1.0.0-SNAPSHOT

At the bottom, a note says: "* Enterprise Features require a valid License Key." To the right, a code editor shows the generated Main.java and pom.xml files:

```
Main.java
public static void main(String... param) {
    DemoApplication app = new DemoApplicationBuilder()
        .withUsername("your-dbms-username")
        .withPassword("your-dbms-password")
        .build();

    // You are ready to go!

    app.stop();
}
```

```
pom.xml
```

Learn More

Trial License? Contact:

Per Minborg

minborg@speedment.com

Carina Dreifeldt

carina@speedment.com



www.speedment.com/initializer



github.com/speedment/speedment

