ParStream – Turning Data Into Knowledge

White Paper – November 2010

Table of Contents

1. Challenges in Mass-Data Analysis 3

2. Setbacks in Current Database Architecture 4

3. Future Development of Processor Architecture 5

4. ParStream's Approach 6

5. ParStream Architecture 8

5.1. Overview 8

5.2. Index Structure 9

5.3. Query Processing 10

5.4. Scalability 11

5.5. Reliability 13

5.6. Interfaces 13

5.7. Data Import 14

5.8. Supported Platforms 14

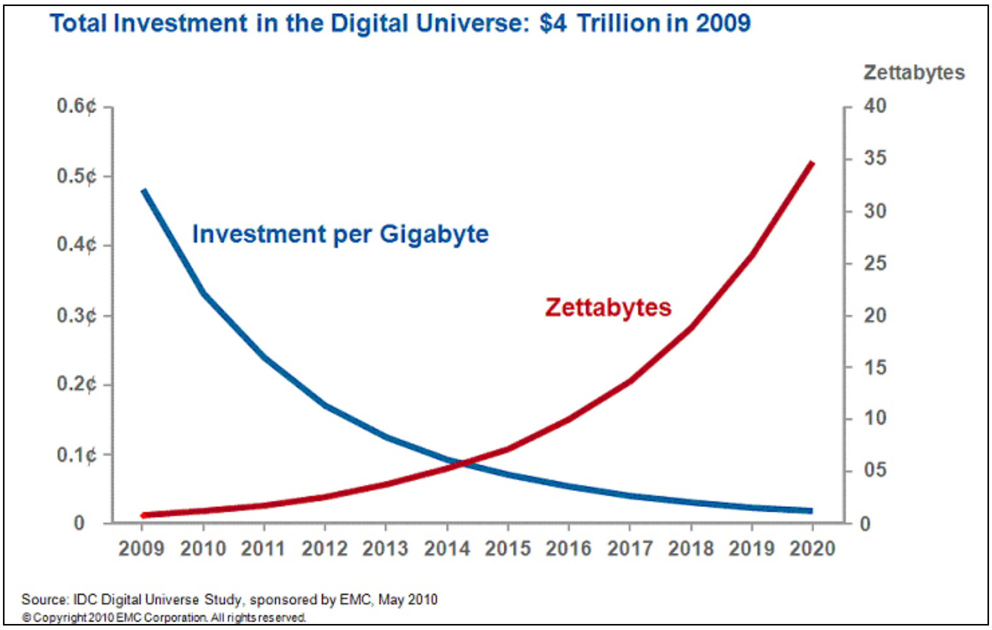
6. Results/Pilots/Performance 15

7. Testimonials 16

1. Challenges in Mass-Data Analysis

The amount of digital data resources has exploded in the past decade. The 2010 IDC Digital Universe Study shows that compared to 2008, the digital universe has grown by 62% or up to 800,000 petabytes (0.8 zettabyte) in 2009.

By the year 2020, IDC predicts the amount of data to be 44 times as big as it was in 2009, thus reaching approximately 35 zettabytes.



Total Investment in the Digital Universe: $4 Trillion in 2009

Source: IDC Digital Universe Study, sponsored by EMC, May 2010
©Copyright 2010 EMC Corporation. All rights reserved.

Figure 1: Trends in Data Volume and Investment per gigabyte

As a result of the exploding amount of data, the demand for applications used for searching and analyzing large datasets (e.g. price comparison, call data records, financial transactions etc.) will significantly grow in all industries, for example:

- **eCommerce -** Web analytics, SEO, affiliate nets, etc.
- **Social Networks -** Ad serving, profiling, targeting, etc.
- **Finance -** Algo trading, trend analysis, fraud detection, etc.
- **Telco -** Profiling, targeting, billing, etc.
- **Energy -** Smart metering, smart grids, wind parks, etc.

However, today's applications require large and expensive infrastructures which consume vast amounts of energy and resources, creating challenges in the analysis of mass-data. In the future, increasingly more terabyte-scale datasets will be used for research, analysis and diagnosis bringing about further difficulties.

2. Setbacks in Current Database Architecture

Current databases are not engineered for mass data, but rather for small data volumes up to 100 million records. Today's databases use outdated, 20-30 year old architectures and the data and index structures are not constructed for efficient analysis of such data volumes. And, because these databases employ sequential algorithms, they are not able to exploit the potential of parallel hardware.

"Existing database architectures are 20-30 years old and are not able to cope with current data sizes."
Conclusion after visiting the *VLDB2010* in Singapore

Algorithmic procedures for indexing large amounts of data have seen relatively few innovations in the past years and decades. Due to the ever-growing amounts of data to be processed, there are rising challenges that traditional database systems cannot cope with. Currently, new and innovative approaches to solve these problems are developed and evaluated. However, some of these approaches seem to be heading in the wrong direction:

"MapReduce isn't suited to calculations that need to occur in near real-time. You can't do anything that takes a relatively short amount of time, so we got rid of it."
Eisar Lipkovitz, Senior Director of Engineering, *Google* in 2010

3. Future Development of Processor Architecture

CPUs are close to reaching their physical speed limits with respect to cycle time. Over the last decades, the performance of processors with a calculating engine in accordance with Moore's law have double each year. From now on, only lower percentage increases are expected.

Future performance increases are due primarily to the concurrent use of multiple processors on a single chip. The cutting edge of this technology are modern graphic cards: Manufacturers are able to integrate hundreds of processor cores on one chip and pair it with ultra-fast memory.

In recent years, such specialized graphic processors have become increasingly more universal as parallel, high-performance computers and are being used not only for 3d graphics applications, but for scientific simulations. Computers with special cards are also known as "desktop supercomputers". Nevertheless many-core systems such as GPUs must be used even more efficiently.

NVIDIA—the leader in manufacturing graphic cards—launched its product line, Fermi, in 2009. Fermi has memory ECC (error-correcting codes) which can correct single bit errors automatically. This is an important prerequisite for the use of graphic cards in business-critical applications because ECC capability is an absolute must for many companies.

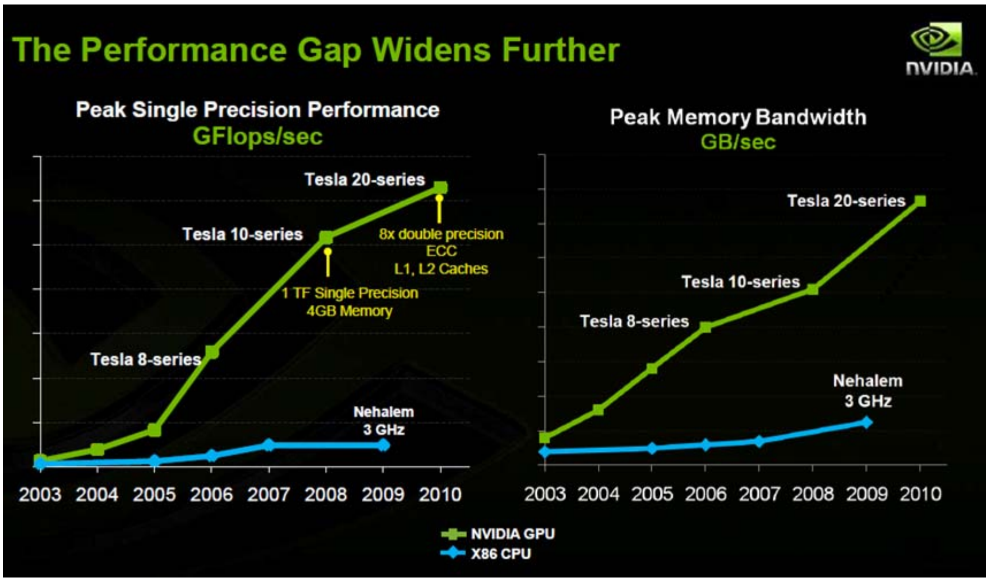


Figure 2: GPU and CPU technology from 2003 - 2010

4. ParStream's Approach

ParStream offers a revolutionary approach to high-performance data analysis. It addresses the problems arising from rapidly increasing data volumes in modern business, as well as scientific application scenarios.

ParStream …

- has a unique index technology.
- comes with efficient parallel processing.
- exploits the parallel power of GPUs.
- is ultra-fast, even with billions of records.
- scales linearly up to petabytes.
- offers real-time analysis and continuous import.
- is cost and energy efficient.

ParStream uses an unique indexing technology which enables efficient parallel processing on parallel architectures. Hardware and energy costs are substantially reduced while overall performance is optimized, thanks to ParStream's use of GPUs—in addition to CPUs—to index and execute queries.

Close-to-real-time analysis is obtained through simultaneous importing, indexing and querying of data. The developers of ParStream strive to continuously improve on the product by working closely with universities, research partners, and clients.

ParStream is the right choice when…

- extreme amounts of data are to be searched and filtered.
- filters use many columns in various combinations (ad-hoc queries).
- complex queries are performed frequently.
- datasets are continuously growing.
- close-to-real-time analytical results are expected.
- infrastructure and operating costs need to be optimized.

5. ParStream Architecture

5.1. Overview

The technical architecture of the ParStream database can be roughly divided into the following areas (see Figure 3):

- During the extract, transform and load (*ETL*) process, the supplied data is read and processed so that it can be forwarded to the actual loader.

- In the next step, ParStream generates and stores necessary index structures and data representations. Input data can be stored in row and/or column-oriented data stores. Here configurable optimizations regarding sorting, compression and partitioning are accounted for.

- Once data is loaded into the server, the user can pose queries over a standard interface (SQL) to the database engine. A parser then interprets the queries and a query executor executes it in parallel.

- The optimizer is used to generate the initial query plan starting from the declarative request. This initial execution plan is used to start processing. However, this initial plan can be altered dynamically during runtime, e.g., changing the level of parallelism.

- The query executor also makes optimal use of the available infrastructure. Parts of the query plan are executed on the GPU, other parts on the CPU. The right balance between GPU and CPU usage is also analysed and altered at runtime.

- All available parts of the query exploit bitmap information where possible. Logical filter conditions and aggregations can thus be calculated and forwarded to the client by highly efficient bitmap operations.

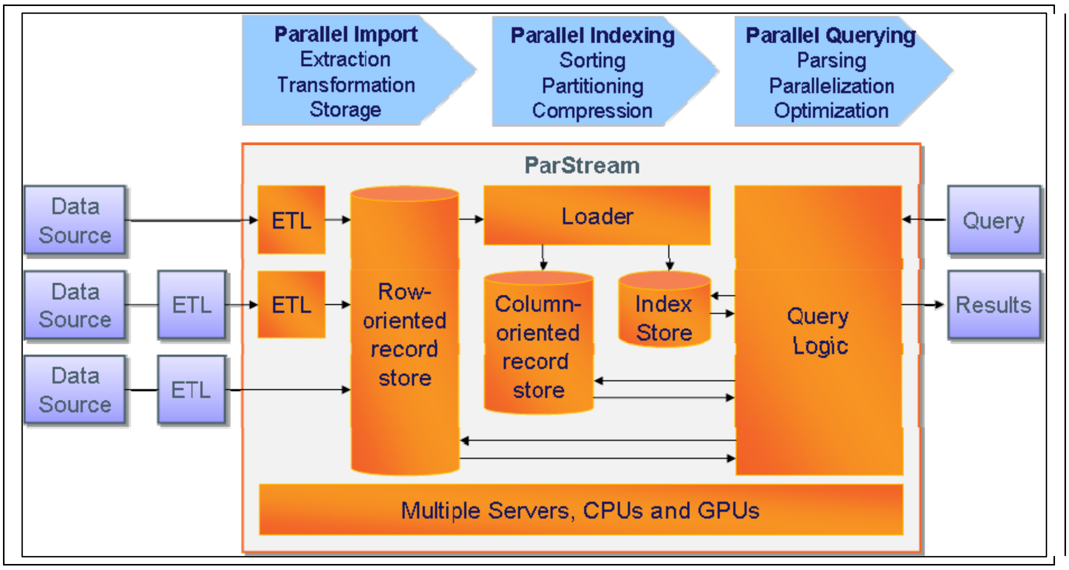


Figure 3: ParStream Architecture

5.2. Index Structure

ParStream's innovative index structure enables efficient parallel processing, allowing for unsurpassed levels of performance. Key features of the underlying database engine include:

- A column-oriented bitmap index using a unique data structure.
- A data structure that allows allows processing in compressed form.
- No need for index decompression as in other database and indexing systems.

Use of ParStream and its highly efficient query engine yield significant advantages over regular database systems. ParStream offers:

- faster index operations,
- shorter response times,
- substantially less CPU-load,
- efficient parallel processing of search and analysis,
- capability of adding new data to the index during query execution,
- close-to-real-time importing and querying of data.

5.3. Query Processing

Query processing inside ParStream is based around the concept of query nodes. A query node represents a specific operation, e.g., filter or aggregation nodes.

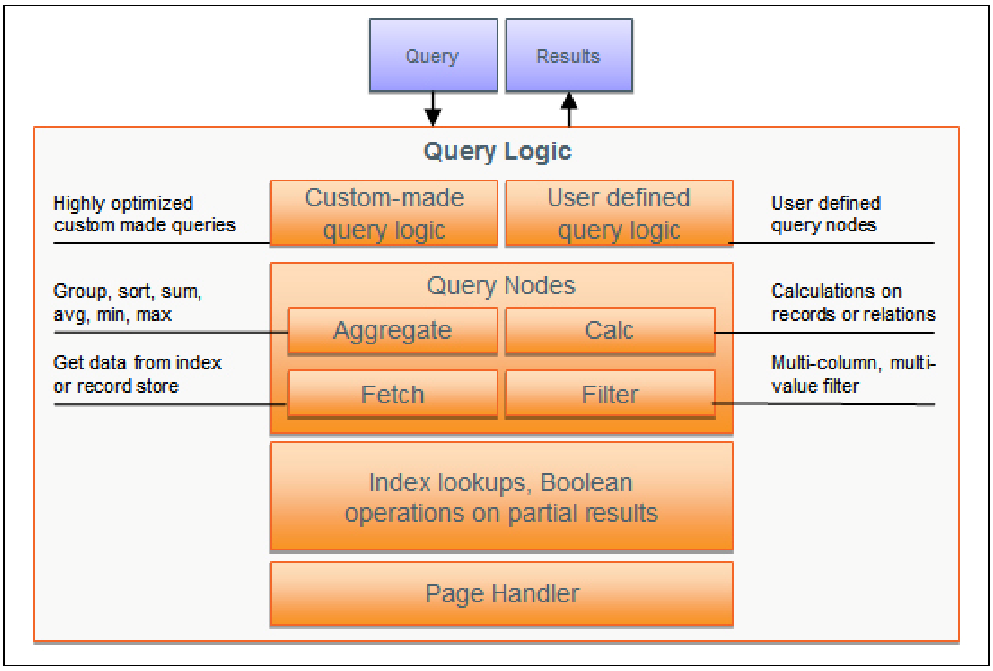

Query
Results
Query Logic
Highly optimized custom made queries
Custom-made query logic
User defined query logic
User defined query nodes
Group, sort, sum, avg, min, max
Query Nodes
Aggregate
Calc
Calculations on records or relations
Get data from index or record store
Fetch
Filter
Multi-column, multi-value filter
Index lookups, Boolean operations on partial results
Page Handler

Figure 4: ParStream's query processing

The entire flow of data through the processing engine is represented by a number of query nodes that form a tree. The fetch nodes are found at the bottom of the tree. Above the fetch level, various operations are performed on the data by the query nodes. This can include common tasks like aggregation, filtering, calculations and sorting. A special query node transports data over network connections. This enables single queries to be distributed over multiple ParStream instances.

On the fetch level, ParStream provides several uses all available information to minimize the I/O load of the actual data fetching. This process makes heavily use of filter conditions and index structures.

The query processing engine automatically analyzes data dependencies and parallelizes relevant parts of the query tree. This allows the execution engine to execute all query nodes at maximum throughput and minimum latency. The entire process is completely transparent to both users and developers.

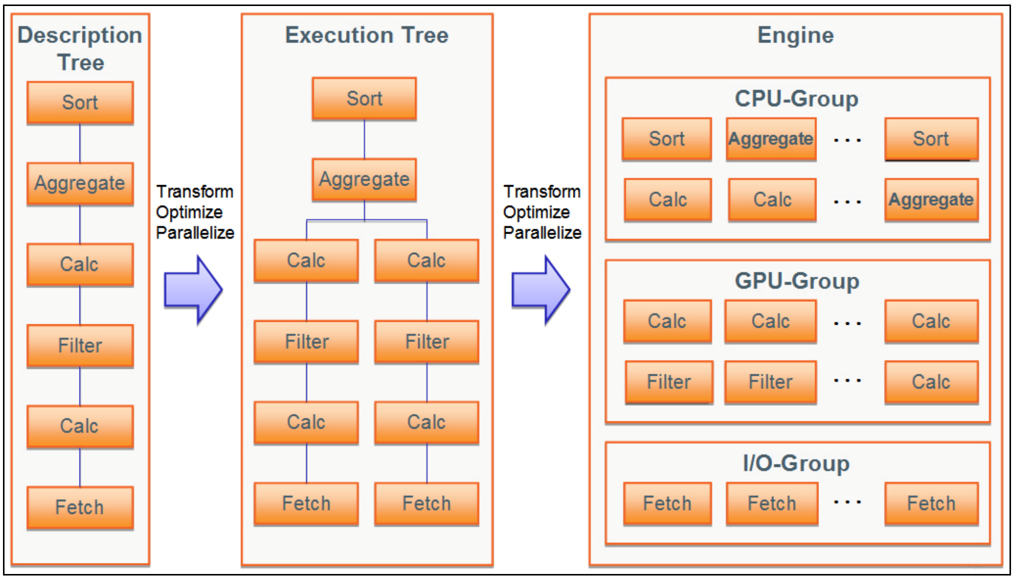


Figure 5: Tree transformation and optimization

ParStream not only supports all previously mentioned database operations, but also features specialized tasks like cluster analysis. Such special query nodes can be developed with ParStream's powerful and flexible API. Thus, developers can write their own specialized query nodes and plug these into the execution engine.

These custom nodes, enable the development of customer-specific solutions. These can be implemented with minimum effort and directly benefit from the power of ParStream and its transparent optimization and parallelization.

5.4. Scalability

ParStream can be scaled in all directions, depending on the needs of the user:

- Scale Up
- Scale Out
- Optimized use of Resources

Scale Up

A single node can contain anything from a single CPU socket, no GPU card, up to 4 CPU sockets or 8 GPU cards. All TCP/IP capable node interconnects are possible, as well as mixed node configurations, allowing queries to be executed on the best possible hardware for a reasonable price.

All data can be stored and distributed on local disc arrays or on a centralized storage system depending on performance needs, reliability and ease of administration.

Scale Out

Multiple nodes perform the index processing completely independently. Only the minimal possible result sets are transferred for further processing.

Additional nodes can be added during run-time for scaling up to meet new performance requirements. Data is load balanced over the nodes at run-time. Multiple data copies can be stored on different nodes to increase the availability.

Rebalancing with no or minimal downtime (depending on different factors) is possible.

Figure 6: ParStream Server Rack

Optimized use of Resources

Inside one single processing node, as well as inside single data partitions, query processing can be parallelized to achieve minimal response time, by using all available resources (CPU, GPU, IO- Channels).

5.5. Reliability

The reliability of a ParStream database is guaranteed through several product features and fully supports multiple-server environments. The ParStream database allows the usage of an arbitrary number of servers. Each server can be configured to replicate the entire or only parts of the data store.

Several load-balancing algorithms will pass a complete query to one of the servers or parts of one query to several, even redundant servers. The query can be sent to any of the cluster members, allowing customers to use a load-balancing and fail-over configuration according to their own need, e.g. round robin query distribution.

To enable the full performance, modern graphic cards are supported to substantially decrease the query response time. Each server can contain a number of NVIDIA Tesla Computing Processors to take advantage of multi-core supercomputing.

The combination of a multi-server environment and the high-performance computing capabilities of a NVIDIA Tesla Computing Processor will strongly increase the reliability and scalability of the ParStream database system.

Hybrid scenarios use GPU boards for best performance on most used index data and use CPU processor for less frequently used data. This will result in optimal performance and cost balance for our customers.

5.6. Interfaces

The ParStream server can be queried using any of the following three methods:

- At a high level, we provide a JDBC driver which enables the implementation of a cross platform front end. This allows ParStream to be used with any application capable of using a standard JDBC interface. For example, a Java applet can be built. This applet can be used to query the ParStream server from within any web browser.

- At mid-level, queries can be submitted as SQL code. Additionally, other descriptive query implementations are available, e.g., a JSON format. This allows the user to define queries, which cannot easily be expressed in standard SQL. The results are then sent back to the client as CSV text or in binary format.

- At a low level, ParStream's C++ API and base classes can be used to write user defined query nodes that are stored in dynamic libraries. A developer can thus integrate his own query nodes into a tree description and register it dynamically into the ParStream server. Such user-defined queries can be executed via a TCP/IP connection and are also integrated into ParStream's parallel execution framework. This interface layer allows the formulation of queries that cannot be expressed using SQL.

5.7. Data Import

One of ParStream’s strengths is its ability to import CSV files at unprecedented speeds. This is based on two factors:

First of all, the index is much faster in adding data than indexes used in most other databases. Secondly, the importer partitions and sorts the data in parallel, which exploits the capabilities of today’s multi-core processors.

Additionally, the import process may run outside the query process enabling the user to ship the finished data and index files to the servers. In this way, the import’s CPU and I/O-load can be separated and moved to different machines.

Another remarkable feature of ParStream is its ability to optionally operate on a CSV record store instead of column stores. This accelerates the import because only the indexes need to be written. Plus, since one usually wants to save the original CSV files, no additional hard drive memory is wasted.

5.8. Supported Platforms

Currently ParStream is available on a number of Linux Distributions including RedHat Enterprise Linux, Novell Enterprise Linux and Debian Lenny running on X86_64 CPUs. On request, ParStream will be ported to other platforms.

GPU acceleration is available on all platforms which are supported by NVIDIA and ParStream.

The ParStream Appliance is based on CentOS (a RedHat Enterprise Linux derivative) and has full support for GPU acceleration on NVIDIA Fermi Cards.

6. Results / Pilots / Performance

ParStream has proven its performance in several productive scenarios.

The figure below shows a performance comparison of ParStream and DBMS *X* as platform for an online search engine for travel offerings.

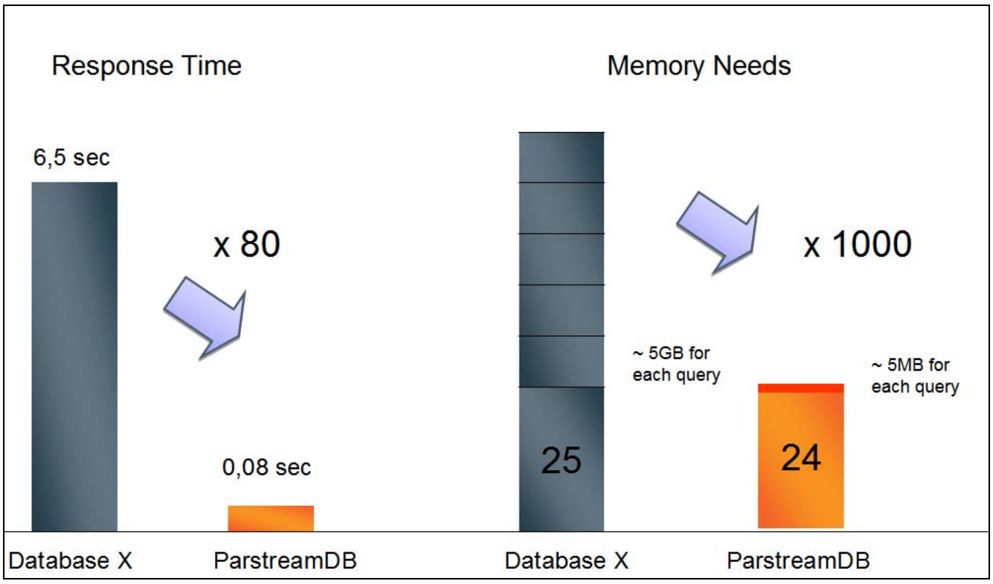


Figure 7: Performance and memory comparison

The task is to deliver 100 travel offers out of 1 billion data records based on 25 independent (optional) filter criteria. ParStream offers a response time 80 times faster than the conventional DBMS *X*. More importantly, the memory requirements for each query are only about 5 megabytes compared to 5 gigabyte for each query in DBMS *X*.

In a productive environment the other database solution needs about 400 servers, whereas ParStream is run on about 20 servers. This leads to an important decrease in infrastructure cost and power consumption.

Some more examples:

Industry	**Challenge**	**Performance results**
Web analytics	Concurrent user calculations	1 billion records in 15 ms
SEO	Data mining	1 billion records joined together in less than 1 second
Market research	Flexible multi-column filtering & grouping	5000 queries, >1000 columns per online-analysis
Climate research	Filter & geo-clustering	3 billion records in 100 ms needed to scale up to 3 petabytes

7. Testimonials

Professor Volker Markl, renowned expert on modern database systems, has received several awards for his work, e.g. IBM Outstanding Technological Achievement Award for the development of learning optimizers for DB2, Hewlett-Packard Open Innovation Award for his work on parallel data processing on clouds:

"ParStream is an extremely innovative idea for processing data in parallel. It exploits the technological trend of using graphic processor units, which are much more powerful than central processing units of modern servers."

"Based on the current state of technology, the technological advancement of empulse in the area of GPU-based databases and the complexity of such an endeavour, it is unlikely that competitors will be able to develop a comparable product within the next two years."

"The fundamental concept of ParStream , as well as its innovative aspects, have convinced me completely. I have decided to support empulse's endeavours actively as a scientific partner, especially in the area of optimization algorithms and components balancing CPU and GPU execution."

A leading web-analytics company and customer of ParStream:

"ParStream reduced our response time from 3 minutes to 15 ms."

"On average, ParStream outperforms competing commercial, column-oriented databases by a factor of 35."

"ParStream scales linearly: The response times grow as expected for large data volumes. No decrease in performance could be observed."

"Over the last few months, ParStream has been running stable in production with multiple instances running on different servers."