The Data Cyclotron: Juggling data and queries for a data warehouse audience
Goncalves, R.

Citation for published version (APA):
Pereira Gonçalves, R. A. (2013). The Data Cyclotron: Juggling data and queries for a data warehouse audience

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Summary

The grand challenge for distributed query processing is to come up with a self-organizing architecture which exploits all hardware resources for the current workload, defines an accurate database subset, minimizes response time, and maximizes throughput without a single point for global coordination.

The Data Cyclotron architecture addresses this grand challenge using turbulent data movement through a storage ring built from distributed main memory. It capitalizes the functionality offered by modern remote-DMA network hardware. Its design reconsiders the network cost model followed by most distributed systems, i.e., slow connections in the past.

Queries assigned to individual nodes interact with the Data Cyclotron by requesting and picking data fragments up from a database subset continuously flowing around, i.e., the *hot-set*. The *hot-set* is dynamically adjusted based on ring characteristics such as load and workload demands.

Composed by the most relevant data for the workload, the *hot-set* is kept as light as possible. Only fragments with a high cumulative query interest over the cycles are kept in the ring. Therefore, a change in the workload focus, i.e., new input data is required, the *hot-set* is re-adjusted. The readjustment is triggered by query requests for data access. The Data Cyclotron’s self-organization in a distributed manner, keeping optimal resource utilization, gradually replaces the data that composes the *hot-set* to accommodate the current workload.

The Data Cyclotron leaves the decision to execute a query to the nodes, i.e., there is no global cost model for query assignment. It makes the system flexible to scale. It exploits the fact that each individual query can be processed at any node; we are sure that the relevant data will pass by. This way, the processing node does not need to be chosen with respect to data locality. Instead, performance factors crucial for the system at large are taken into account. Therefore, the Data Cyclotron, instead of moving the data to the queries (DDBMS) or a query to the data (peer-2-peer), it moves data and queries as two complementary entities. As far we are aware, it is an innovative and
simple strategy rarely, or maybe never, used before in the distributed query processing environments.

The approach is first illustrated using an extensive simulation study. The results underpin the hot-set management robustness in turbulent workload scenarios. Furthermore, a fully functional prototype of the proposed architecture has been integrated with a column-store. The integration was tested within a multi-rack cluster equipped with InfiniBand using both micro benchmarks and high-volume workloads based on the well-known decision support benchmark, TPC-H.\footnote{http://www.tpc.org/tpch/}

The results demonstrate the architecture feasibility and inspired the design of a novel solution for distributed query processing, the DaCyDB. The DaCyDB opens a new vista for modern distributed database architectures with a plethora of new research challenges. It uses the decentralized Data Cyclotron architecture to offer a seamless transition between a scale up solution to a scale out solution, or the best case scenario, have both in harmony. This seamless transition is a pillar to explore both models and achieve different levels of parallelism.

For efficient resource utilization and an absentee of central coordination, the DaCyDB exploits the Data Cyclotron pulsating ring instead of the master-slave model. A pulsating ring adaptively grows and shrinks to continuously seek the optimal number of nodes comprising a ring and guaranties the fastest data flow. All decisions are made locally at each node: which query to execute, when to leave, or when to split. Hence, it is possible to have a mesh of heterogeneous rings, and thus make the architecture flexible and robust to difference workload resource demands. The model exploits the independence and autonomy of each individual where everyone, based on the workload flow, works for the same goal, high throughput.