Distributed Data Cube Analytics

Dimitar Misev, Vlad Merticariu, Alex Dumitru and Peter Baumann
School of Engineering and Computer Science, Jacobs University Bremen

JACOBS UNIVERSITY
Large-Scale Scientific Information Systems Research Group

rasdaman
raster data management

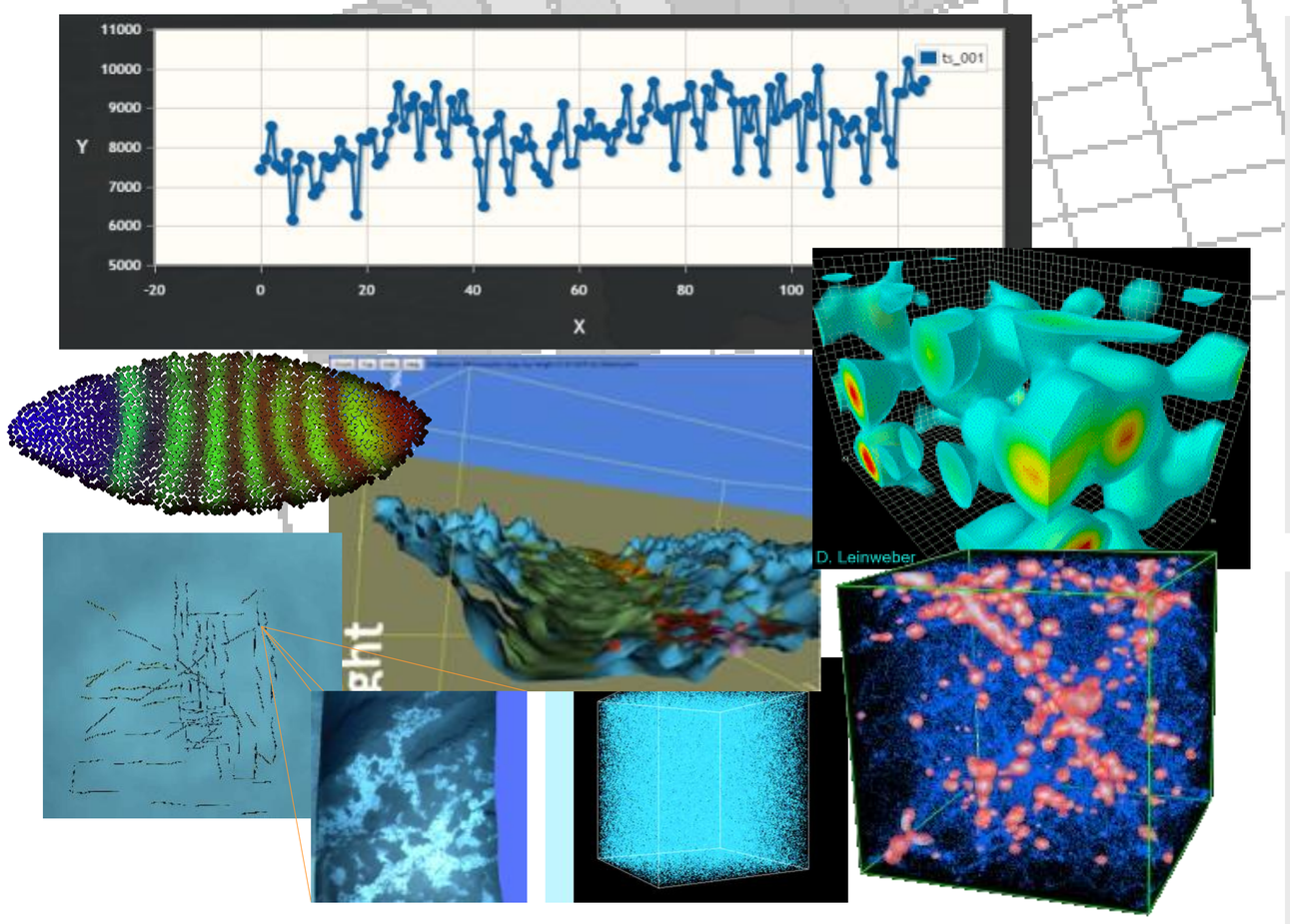

Fig. 1: Examples of n-D array data

1. Introduction

- The *multi-dimensional array data cube model* is becoming increasingly important in a variety of Big Data challenges
- 1-D sensor data, 2-D satellite imagery, 3-D x/y/t image time series as well as x/y/z geophysical voxel data, and 4-D x/y/z/t weather data and astrophysics simulations (Fig. 1) typically appear in massive volumes
- **Array databases** [1,2,3] strive to provide efficient analytics on such data

2. Motivation

- Largest array database installations are well above 100 Terabytes
  - data available is in the order of *Petabytes* and growing
- Query processing on a single machine is impractical with such volumes
  - plus data is distributed across nodes and data centers; moving data to the processing is expensive and often not even possible

3. Distributed processing in rasdaman

- A network of machines (nodes) collaborate in answering a single array query, by automatically distributing its evaluation as independent sub-queries to the peer nodes (Fig. 2)
- *Shared-nothing* architecture based on two main components working together to minimize costly network transfer:
  1. ***Federation Daemon*** handles communication and provides real-time statistics about the data and capabilities of each node
  2. ***Rasdaman Server*** divides work among the appropriate nodes and merges results into a single response

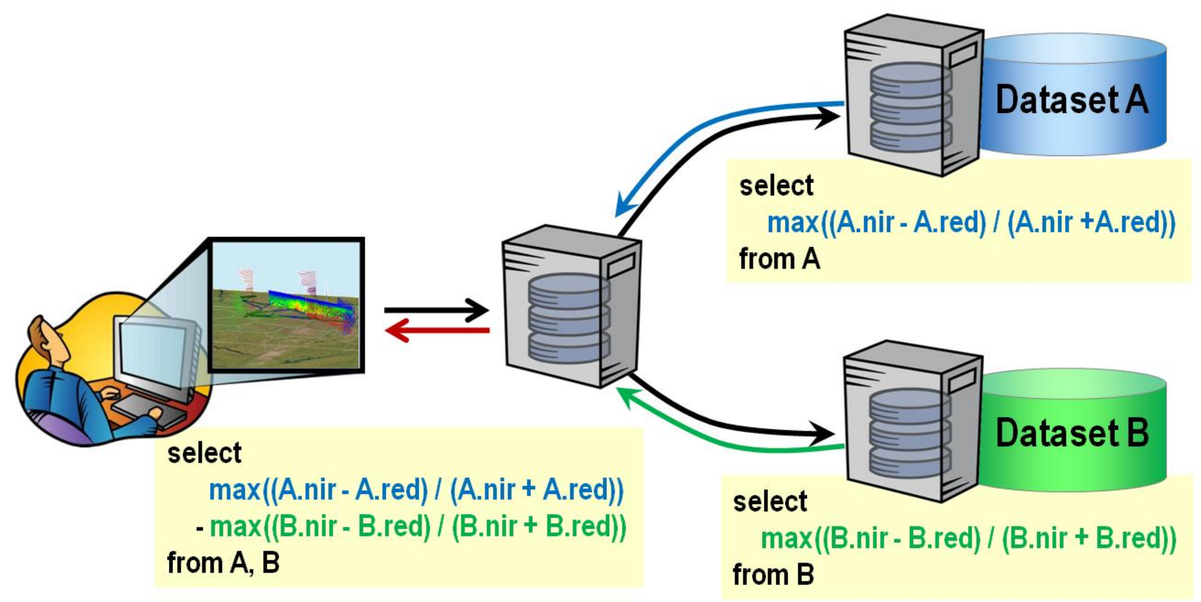

Fig. 2: Example intra-query parallelization

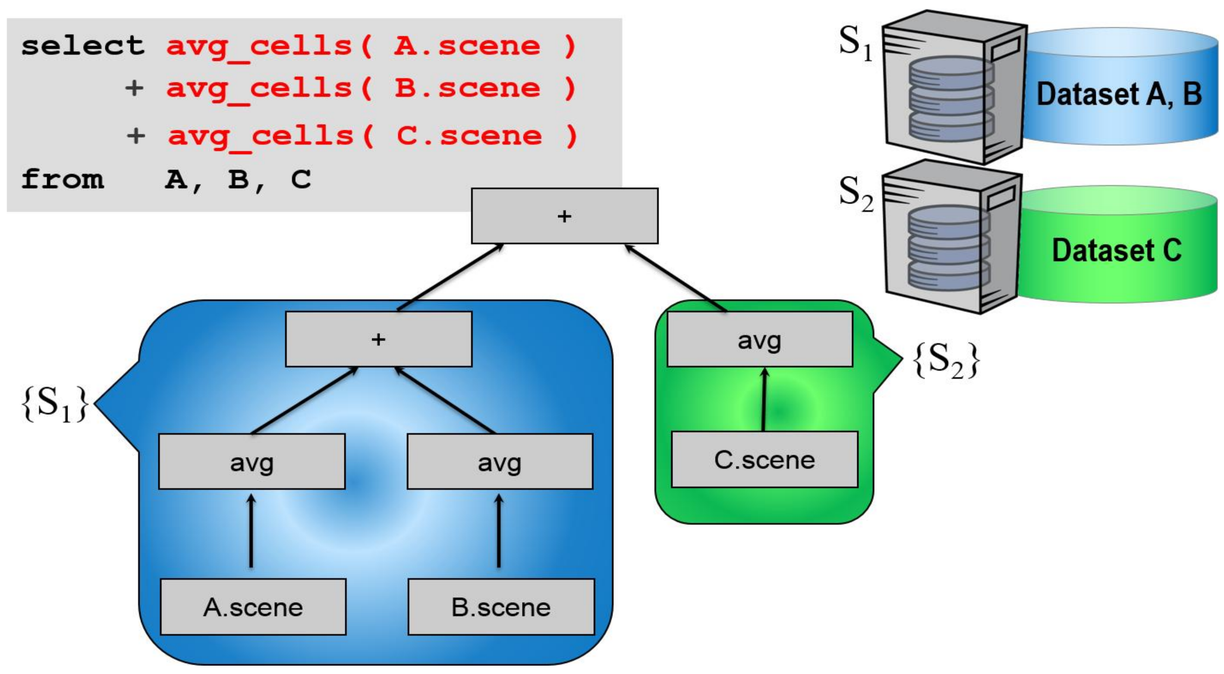

Fig. 3: Query splitting example with query rewriting

4. Query splitting algorithm

To distribute processing the query is **split into sub-queries** (Fig. 3):

1. The query tree is traversed bottom-up from each array object access
2. When tree branches join, node sets that can process each branch are intersected to yield the parent node set
3. The sub-trees represent independent sub-queries when the intersection is an empty set; they are distributed to a node in the node set

5. Conclusion

Our approach has been evaluated in a deployment on *Amazon EC2 nodes (Fig. 5)*. We have deployed an installation on a science cloud hosted by the German giant *T-Systems*. In the *EarthServer-2* project we continue working towards **Petabyte-sized** data cubes distribution.

References and further information: http://rasdaman.org/socc15

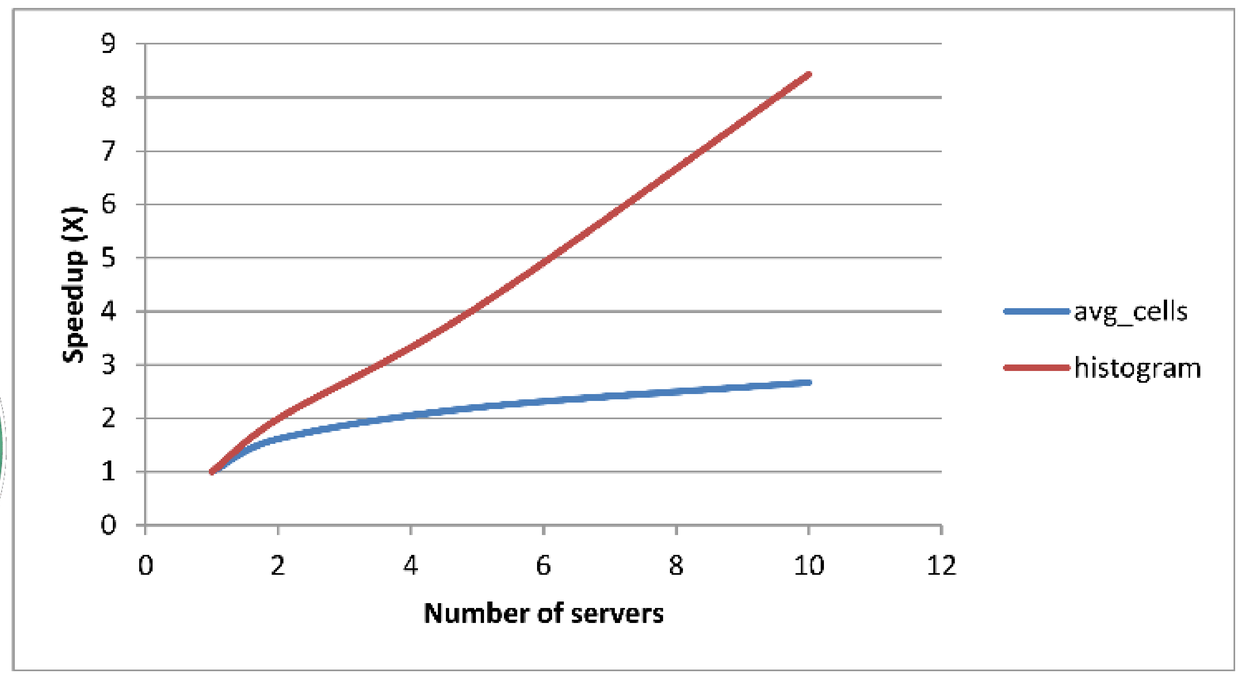

Fig. 5: Speedup on light vs heavy computation