Data-Intensive Systems

Computer Science Day, June 15, 2011

Ira Assent

Data-Intensive Systems

- Behind every successful website and many successful information systems, there is a powerful database!
- Examples:
  - Shipment tracking
  - Web shops
  - Social networks
  - Search engines….
- From traditional databases to diverse systems
- From business data to all data
  - Streaming and sensor data, semi-structured and unstructured data
  - Multidimensional data, temporal data, spatio-temporal data
- Examples
  - Mobile data management
  - Data mining
  - … and many more
  - See also our website at http://cs.au.dk/research/areas/data-intensive-systems/

Staff

- Ira Assent, associate professor
- Christian S. Jensen, professor
- Vaida Ceikute, Ph.D. student
- Xiaohui Li, visiting Ph.D. student

- 7 Ph.D. students, 1-2 postdocs, 1 visiting professor coming this fall




Projects

- Streamspin
  - Enable sites that are for mobile services what YouTube is for video
    - Easy mobile service creation and sharing
    - Advanced spatial and social context functionality
    - Be an open, extensible, and scalable service delivery infrastructure
- MOVE
  - Knowledge extraction from massive data about moving objects
    - Cross-cutting activities, showcases, and evaluation
    - Representation of movement data and spatio-temporal databases
    - Analysis of movement and spatio-temporal data mining
- eData
  - Robust analysis in the context of imperfect data in e-Science
    - Detect and correct anomalies effectively
    - on-line, interactive, lineage-preserving, and semi-automatic
    - Scalable algorithms

Projects (2)

- GEOCROWD
  - Creating a Geospatial Knowledge World:
    - Advance the state-of-the-art in collecting, storing, analyzing, processing, reconciling, and publishing user-generated geospatial information on the Web
- REDUCTION
  - Reducing the environmental footprint of fleets of vehicles
    - Optimizing the behavior of drivers
    - Supporting eco-routing of vehicles
    - Enabling transparency in multi-modal transportation
- WallViz
  - Collaborative analysis, joint decision making on wall-sized displays
    - Scale to massive data collections
    - Support ad-hoc queries
    - Automatically provide entry points for analysis

WallViz

Improving decision making
from massive data collections
using wall-sized, highly interactive visualizations

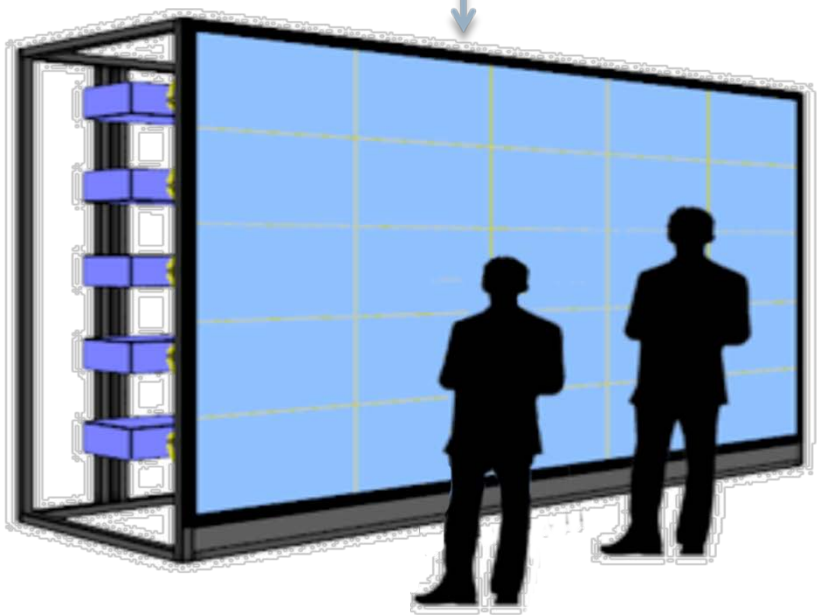
- Decisions are increasingly informed by analysis of massive data collections
- Manual exploration hard;
  automatic analysis infeasible
- Results in information overload,
  poor decisions

- Visualization helps deal with massive data collections

Funded by Strategic Research Council 2011-2014

Setup of the WallViz project

Human-Centered Computing (DIKU)



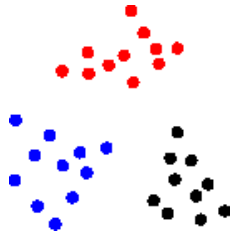
Case partners in health care, finance, sustainability

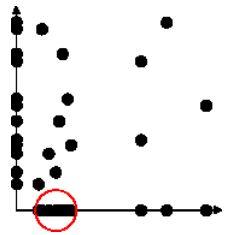
Evaluation, field studies

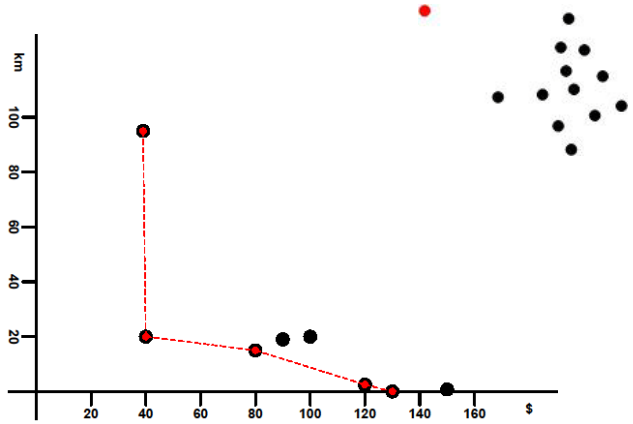
Data management (CS, AU)

Research on data management in WallViz (1)

- Where do users start?
  - Data is overwhelming!
- Data mining and skyline queries to the rescue
  - Create smart entry points for decision making
  - "Good guesses" at what might be interesting in the data
  - Comes in different flavors:
    - Summarize data in groups (clustering), filter out relevant information (subspace clustering)
    - Find the "odd-one-out" (outlier detection)
    - Find all the best options (skyline)





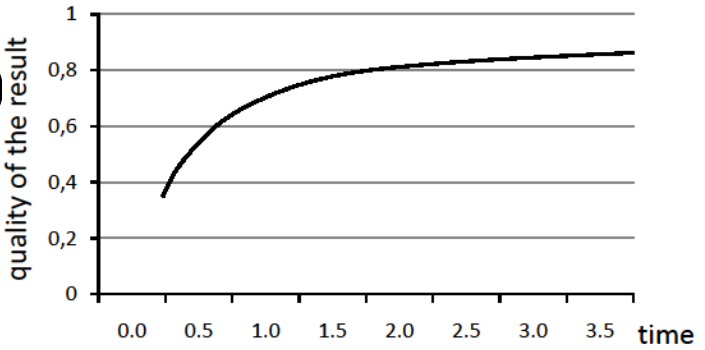


Research on data management in WallViz (2)

- How do users **interact** with the system?
  - Loading all data is too slow!

- Efficient query processing to the rescue
  - Narrow the problem down to what is really needed
    - Pre-compute some of the answers that help in answering most of the questions (pre-materialization)
    - Compute the rest when necessary (just-in-time queries)
    - Select the relevant information that needs to be searched (indexing, subspace analysis)

  - While computing, report to the user what you already know
    - Give a rough answer and keep improving on that (progressive queries, anytime mining)
  - Don't repeat yourself
    - Maintain (partial) results that are likely to be re-used in decision making (caching)



How We Typically Work

- We target some real problem that we find interesting.
- We define the problem precisely.
- We develop a solution that is typically a data structure or an algorithm, i.e., a concrete technique.
- To evaluate, we build prototypes.
  - These are built for the purpose of studying the properties of our solutions.
  - We are often interested in performance, e.g., runtime, space usage, communication cost.
- For some solutions we state formal properties that we then prove, e.g., the correctness of a particular technique

- Brief: isolate and define problem, construct, then evaluate

Interested?

- Come talk to us!

- We currently have open M.Sc. topics

Acknowledgments: slides include material provided by Kasper Hornbæk, DIKU