UNIVERSITY OF COPENHAGEN

Managing Geospatial Big Data: Lessons Learned and Future Perspectives

Marcos Vaz Salles
Associate Professor, University of Copenhagen (DIKU)

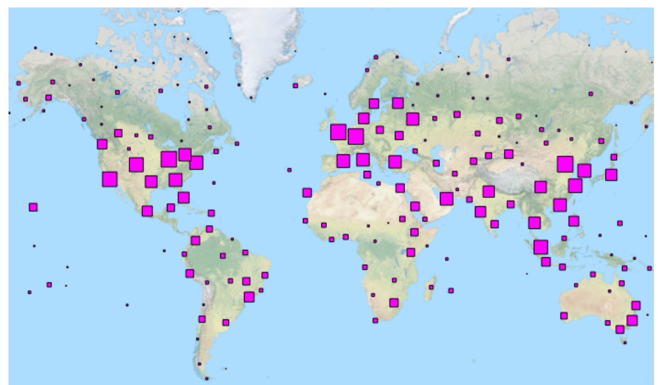
Geospatial Applications

- Notion of geographical space fundamental to human activities and the environment
- Applications with spatial characteristics among the most exciting in computing

https://media.licdn.com/mpr/mpr/jc/AAEAAQAAAAAAAAX8AAAAJDEyMjY2Zjc3LWIyOWMtNDY2YS1hZGE2LWRjNDU3MDg2OTE0ZQ.jpg

https://en.wikipedia.org/wiki/Spatial_analysis#/media/File:Snow-cholera-map.jpg

Location-based services, augmented reality

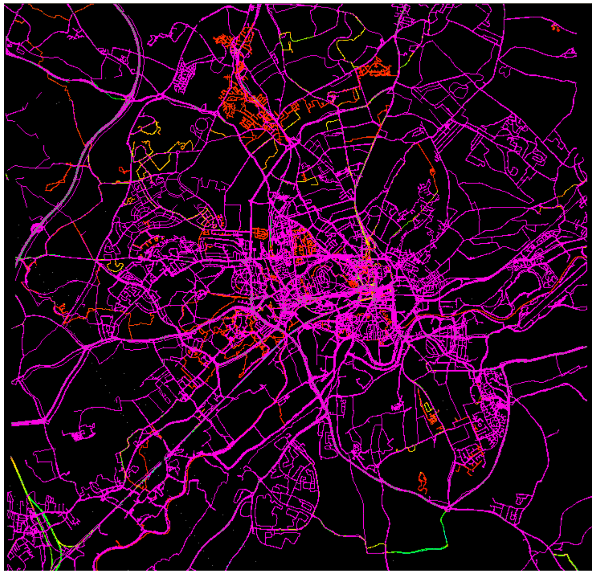


Data visualizations, e.g., maps

Spatial analysis

Challenges in Geospatial Applications

- **Big variation in data formats and volume**
  - Some data is “cheap” to obtain, other is “expensive”
  - Examples: Drone images, GPS traces, satellite data vs. visual ranking of breeds by humans



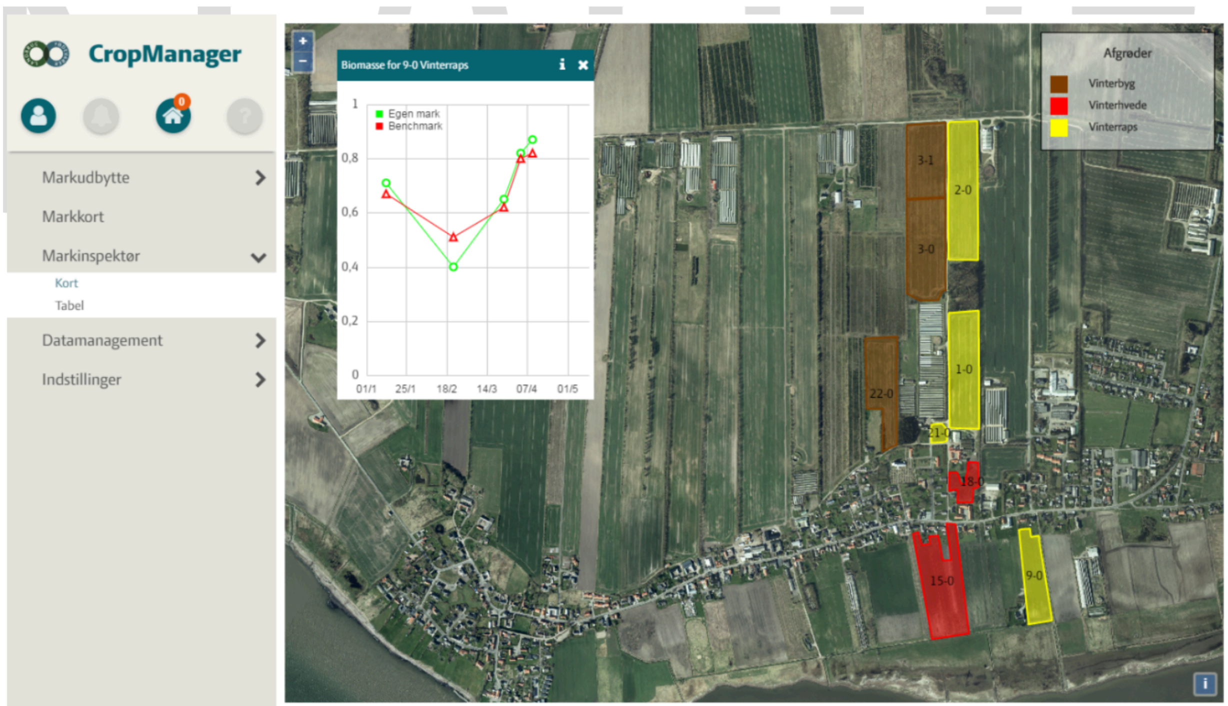
http://wiki.openstreetmap.org/w/images/e/e7/Nottingham_gps_traces_ex_osm_20110105.png

https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle#/media/File:Interspect_UAV_B_3.1.png

http://futurecropping.eng.au.dk/maps/204

Challenges in Geospatial Applications

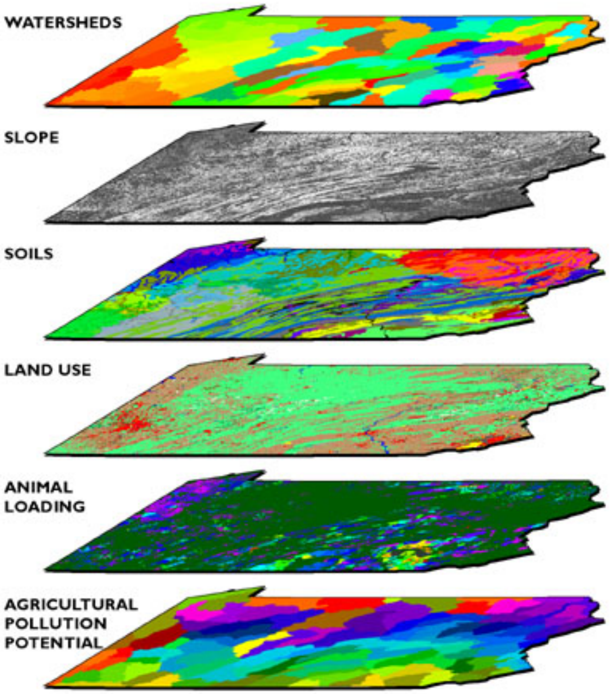
- **Large amount of users and potentially complex simultaneous requests**
  - Popular datasets need to be serviced to many users, and transformed by different programs
  - Examples: Google Maps & shortest paths, Future Cropping data platform & analysis services



NDVI benchmarking in SEGES Crop Manager, tech transfer from Future Cropping project

Challenges in Geospatial Applications

- **Much labor needed to derive knowledge from varied data**
  - "Expensive" data can be too small or too noisy for phenomenon studied, not obvious how to leverage "cheap" data
  - Examples: Soil samples vs. satellite or land use data



https://www.e-education.psu.edu/natureofgeoinfo/c9_p6.html

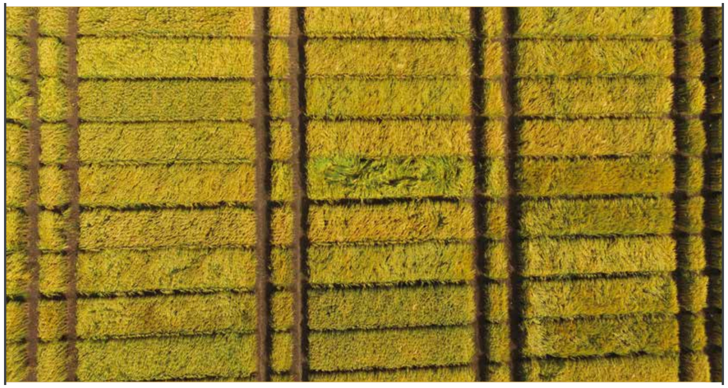
Lessons From Managing Geospatial Data

- **Challenge: Big variation in data formats and volume**
  - **Lesson 1: "Cheap" vs. "expensive" data**
  - **Lesson 2: The rise of standardization, open-source software, and large geospatial datasets**
- **Challenge: Large amount of users and potentially complex simultaneous requests**
  - **Lesson 3: From software to services**
  - **Lesson 4: Telemetry turns behavior into data**
- **Challenge: Much labor needed to derive knowledge from varied data**
  - **Lesson 5: Embed intelligence in services**

Lesson 1: "Cheap" vs. "Expensive" Data

What makes data “expensive”?

http://www.agronomy.k-state.edu/services/soiltesting/farmer-services/soil-analysis/index.html



https://www.nordgen.org/ngdoc/plants/ppp_sekr/PPP_Basic_Documents/Basic_documents/ppp_promoting_nordic_plant_breeding_for_the_future.pdf

- **Lack of automated sensing technology**
  - For example, soil samples or phenotype annotations require human labor *for each sample*
- **Lack of data description (metadata) or data quality controls**
  - For example, drone images can turn out to be very *noisy data* due to measurement errors, including alignment, color filters, variability in cloud cover, variability in RGB profiles across drones

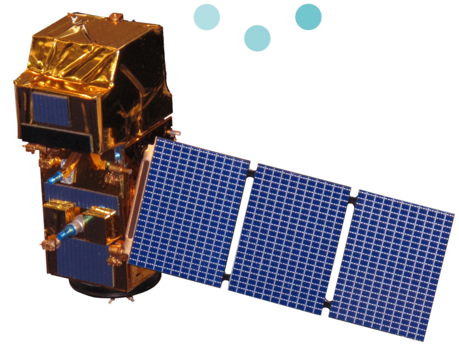
What makes data “expensive”?

https://dribbble.com/shots/2055315-We-Love-Data

- **Lack of data-centric organizational culture, tools, and technology**
  - For example, top management does not see data as first-class entity in business, or there is no data team or platform established
- **Lack of possibilities to externalize data management costs**
  - For example, there may be no service providers in the given area, or data may be strategic differentiator

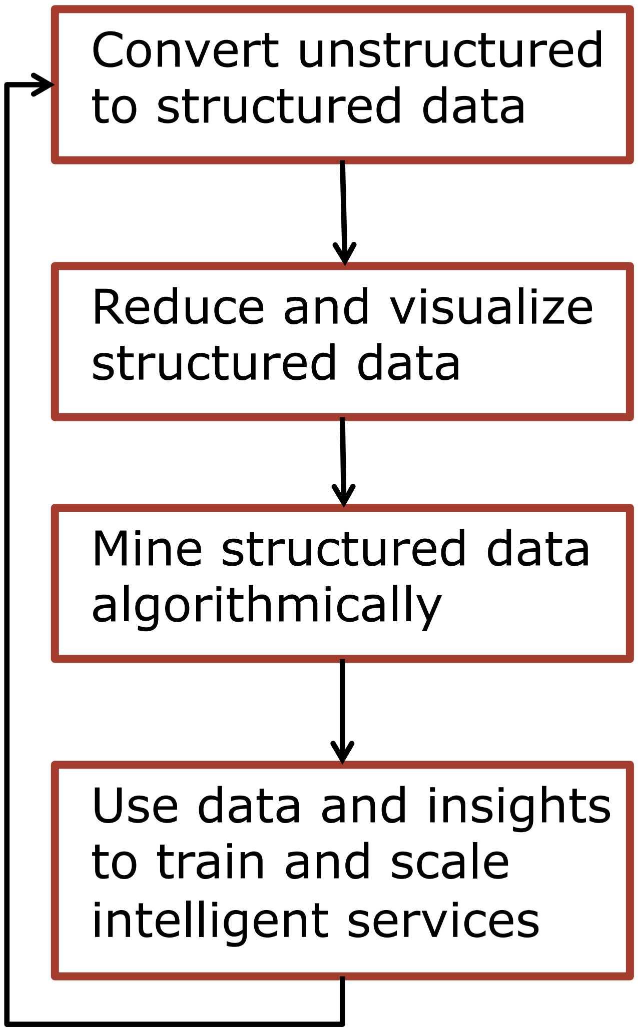
Geodatastyrelsen

http://gst.dk/



Sentinel-2 satellite, image by Rama, https://upload.wikimedia.org/wikipedia/commons/3/3d/Sentinel_2-IMG_5873-white_%28crop%29.jpg

Becoming Data-Centric



- **Every step is important**: Partial achievement is possible
- Achieving the whole loop requires teams of both **data engineers** and **data scientists**

The challenges of exploiting "cheap" data

- Data as **intangible asset**
  - Investment needed to deliver value!
- Getting to **structure**: Goal is to represent data as **tables** (preferred) or **matrices**
  - **Non-trivial transformations** to structure data, e.g., how to think of free text or images as tables or matrices?
  - **Heterogeneity** in representation of data across different databases in different table formats (**schema-level**) or of same data in different sources with different attributes (**instance-level**)
  - **Errors** in data leading to the need for data quality procedures and **data cleaning**

Lesson 2: The Rise of Standardization, Open-Source Software, and Large Geospatial Datasets

Standardization in Geospatial Data

- **Open Geospatial Consortium (OGC)**
  - Standards body for geospatial data
  - Work on data formats, e.g., well known text (WKT), NetCDF
  - Work on protocols, e.g., WFS, WCS, WMS, WPS
- **Open-source software**
  - *Spatial application server and CMS*: GeoServer, GeoNode
  - *Spatial relational database*: PostgreSQL/PostGIS
  - *Raster analytics*: Rasdaman
  - *Spatial Big Data*: Spatial Hadoop, Simba, GeoMesa, GeoWave

http://www.opengeospatial.org/

http://www.osgeo.org/

http://spatialhadoop.cs.umn.edu/

http://www.rasdaman.org/

Standardization helps in structuring data

- **Common data model** eases non-trivial transformations, provides way to leverage previous efforts
- **Open-source software** reduce data heterogeneity at both schema and instance levels
- **Standard formats** allow for large, high-quality datasets to be curated and shared

- **BUT...**
  - Standardization tends to work when data targeted is supposed to become *commonly used across industry*
  - *Spreads costs* of integrating data among participants
  - Hard to achieve when data provides *proprietary competitive advantage* to organizations

The New Problem: Too Much Data!

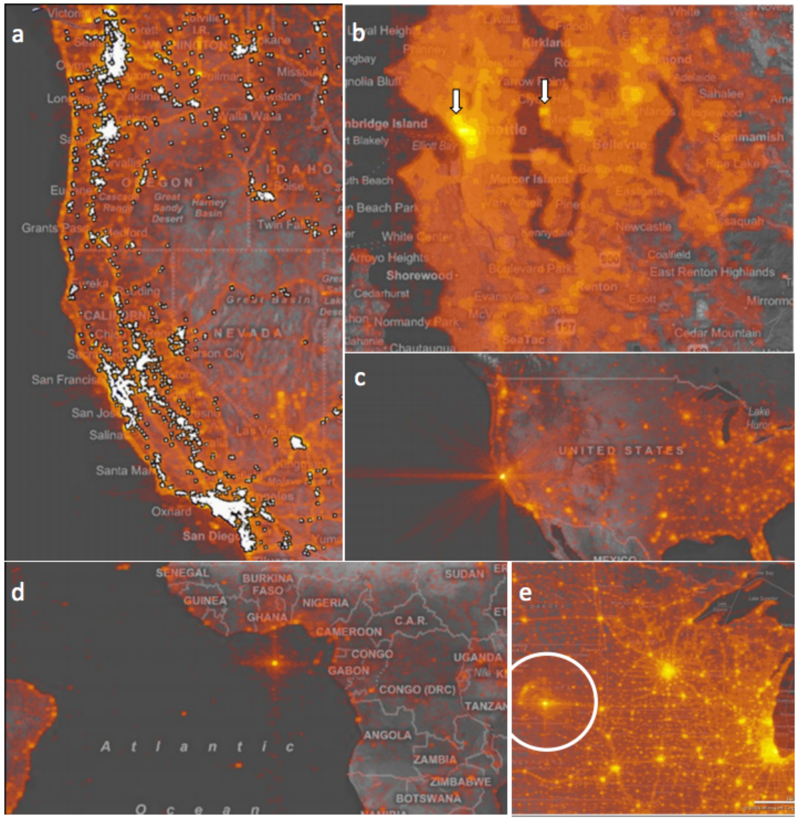
- Standards and open-source software pave the way to large datasets
- **Concern: What if the data does not fit my spreadsheet?**

Data provided by Rasmus L. Hjortshøj at Sejet

Winterbarley 2016				LOCATION ->	Sejet			Fyn			Sj	Fy	Mean	Hildesheim				Sj	Sj	Sj	Sj	Sj	Sj	Sj	Fyn	Sj	Sj	Fyn
Expt	Entry	name	Baymv	pedigree	HKG	RK	CV	HKG	RK	CV	REL	REL	REL-MEAN	HKG	RK	CV	REL	ear emerg. Day in year	straw length cm.	Lodging1	Lodging2	HLV	Protein	Meldg 28/5	Meldug 24/5	Skoldplet_2/5	Sloldplet 28/5	Skoldplet 2/5
C	1	HEJMDAL	modt	apropos/sj 047422	94.9	4	2.5	59.4	6	24.5	105	100	102	86.7	37	6.2	95	148	83	1	6	63.8	9.5	4	4	0	0	0
C	2	PADURA	modt	zephyr//alibi/chess	84.6	37	2.6	49.8	40	6.7	93	84	89	93.4	21	2.0	103	147	80	1	5	65.4	9.9	6	5	3	7	1
C	3	MATROS	modt	himalaya//carat/chess	86.6	31	4.0	51.1	36	9.7	96	86	91	93.6	20	3.3	103	146	90	1	7	65.1	10.1	2	4	1	0	0
C	4	KATHMANDU	rym4	retriever///annerose//himalaya/anisette	84.8	36	4.2	56.0	24	19.5	94	94	94	90.8	27	5.1	100	141	72	0	5	61.5	9.4	5	3	2	4	1
C	5	CONCORDIA	-	sj 047422/sj 053099	88.4	26	6.5	57.7	13	17.2	98	97	97	97.3	4	3.1	107	139	69	0	6	63.4	9.7	4	3	3	8	4
C	6	TALISMAN	rym4	retriever/flagon	86.2	33	3.4	55.5	25	7.4	95	93	94	91.1	26	0.4	100	147	87	0	7	63.6	9.3	3	1	1	3	3
C	7	FRIGG	modt	apropos///annerose//himalaya/anisette	95.1	3	0.3	58.5	11	21.3	105	98	102	92.5	24	2.5	102	147	80	0	7	62.5	9.4	2	3	0	0	2
C	8	BLANDING' 16	-		92.1	13	3.8	59.0	9	9.2	102	99	100	86.2	39	6.9	95	147	84	0	6	63.4	9.9	6	3	1	1	2
C	9	SOBELL	rym4	augusta//kws cassia/matros	90.9	17	4.7	54.5	27	24.5	101	91	96	87.5	35	5.5	96	145	77	1	7	62.0	9.6	3	4	0	2	2
C	10	NEPTUN	modt	sandra/matros	95.1	2	0.4	57.7	14	30.0	105	97	101	89.0	30	5.3	98	147	87	1	4	65.5	9.4	6	4	2	5	1
C	11	C-11	rym4		92.2	11	2.7	51.7	34	11.7	102	87	94	93.6	19	4.5	103	146	80	1	3	66.0	9.6	3	1	0	0	0
C	12	C-12	rym4		79.1	41	2.7	52.8	30	17.6	87	89	88	95.7	8	4.5	105	146	80	2	9	62.0	9.8	3	2	0	0	0
C	13	C-13	rym4		83.9	39	1.9	50.9	39	15.5	93	86	89	98.2	2	3.4	108	145	92	2	7	66.4	10.1	3	1	0	3	1
C	14	C-14	rym4		95.4	1	1.6	49.8	41	14.3	105	84	95	97.3	5	3.7	107	147	81	0	4	65.9	9.7	7	3	1	0	1
C	15	C-15	modt		92.2	12	1.8	57.2	17	15.1	102	96	99	88.9	31	5.9	98	146	84	0	4	63.2	9.5	1	0	0	1	1
C	16	C-16	rym4		88.5	25	3.6	61.2	3	7.0	98	103	100	92.0	25	5.5	101	148	80	1	6	63.5	9.2	6	2	2	1	4
C	17	ZOPHIA	rym4	admiral/daniela	84.5	38	2.7	60.3	4	7.3	93	101	97	95.1	11	4.0	104	148	82	0	4	68.5	9.8	4	4	0	0	1
C	18	C-18	rym4		93.0	5	0.5	59.1	7	25.0	103	99	101	90.7	28	0.1	100	146	75	1	7	65.2	9.3	3	1	0	1	2
C	19	BALTAZAR	rym4	sj 2848//admiral/daniela	90.4	18	3.5	52.2	32	15.0	100	88	94	93.3	22	3.2	102	145	82	1	5	65.4	8.9	1	1	0	3	4
C	20	C-20	-		86.3	32	6.0	58.9	10	11.5	95	99	97	87.9	33	4.7	96	146	83	2	8	62.0	10.1	3	3	0	2	0

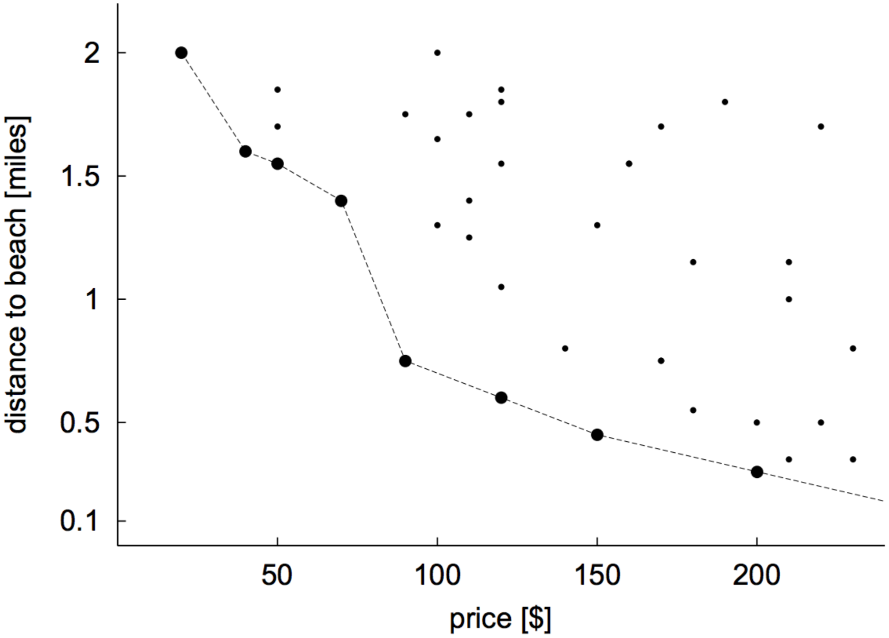
Data Reduction Methods

- **Data aggregation, e.g., heatmaps**



"Hotmap: Looking at Geographic Attention", Danyel Fisher, Microsoft Research (2007)

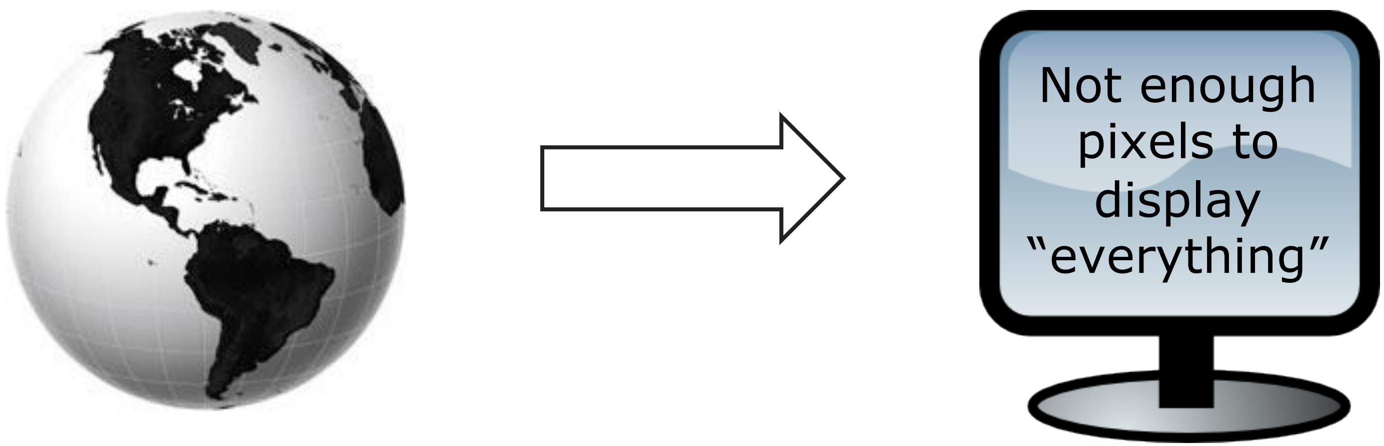
- **Data selection, e.g., skyline, cartographic generalization**



"The Skyline Operator", Stephan Börzsönyi, Donald Kossmann, Konrad Stocker, ICDE 2001

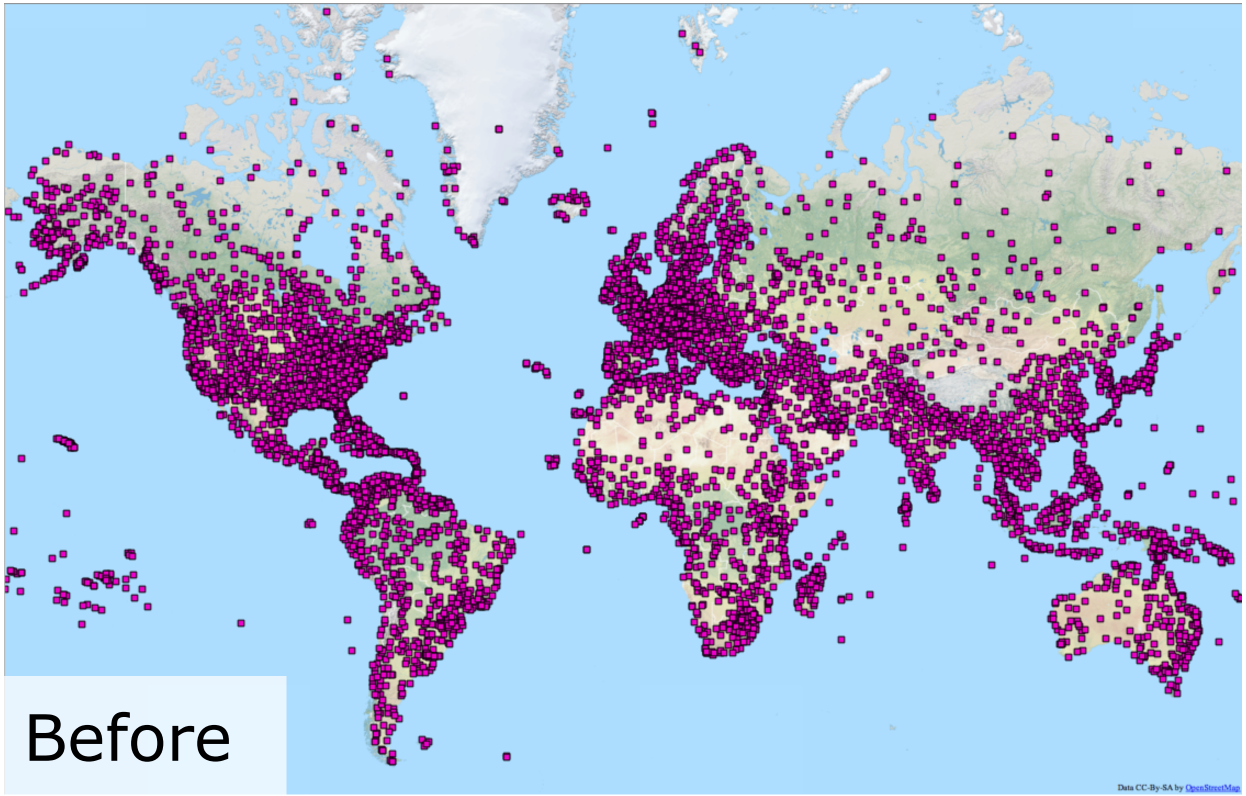
Research Highlight: Declarative Cartography

Adapting data to scale of visualization medium



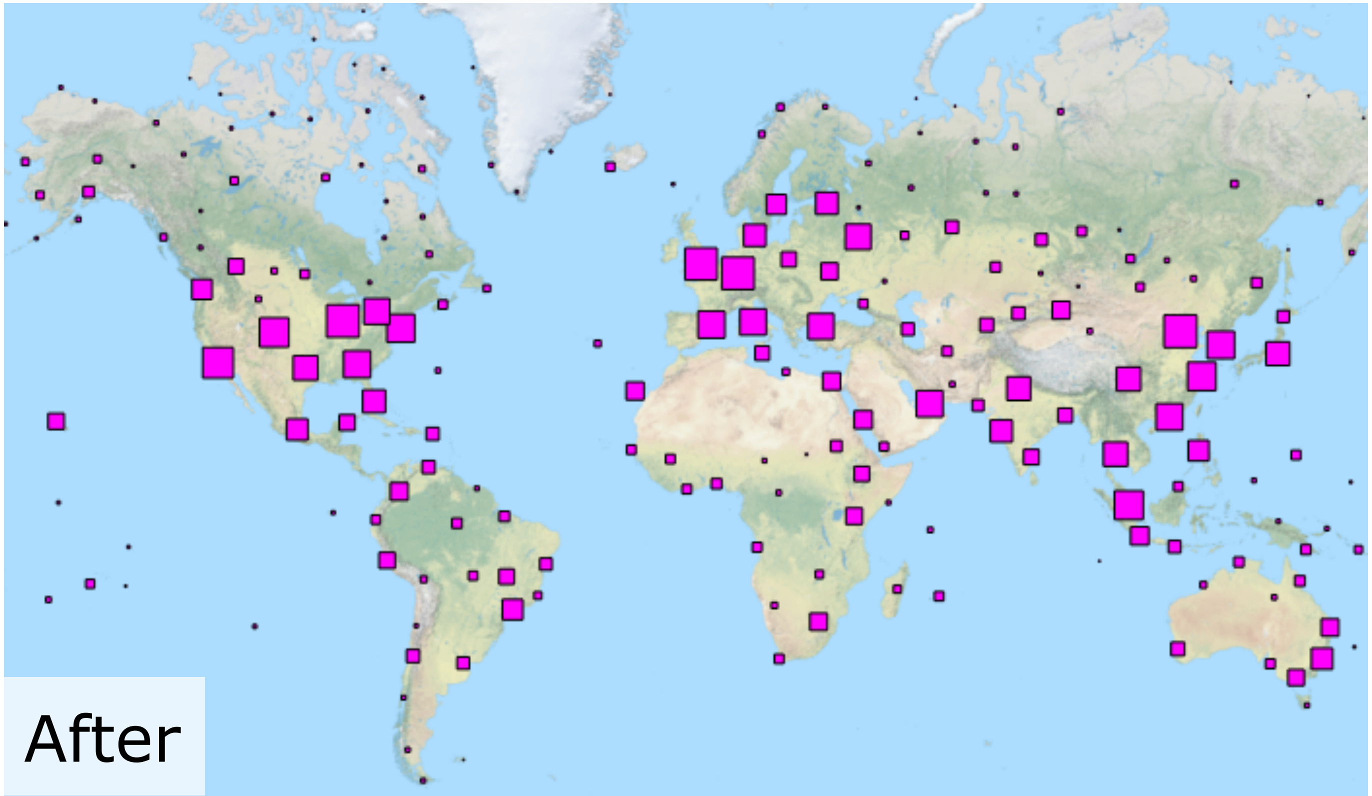
Work done in collaboration with P. K. Kefaloukos
and M. Zachariasen, results in ICDE 2014

Example: Selecting Airports



- Too much information → illegible map
- Not clear how to deal with zooming
- Not obvious how to pick objects to display

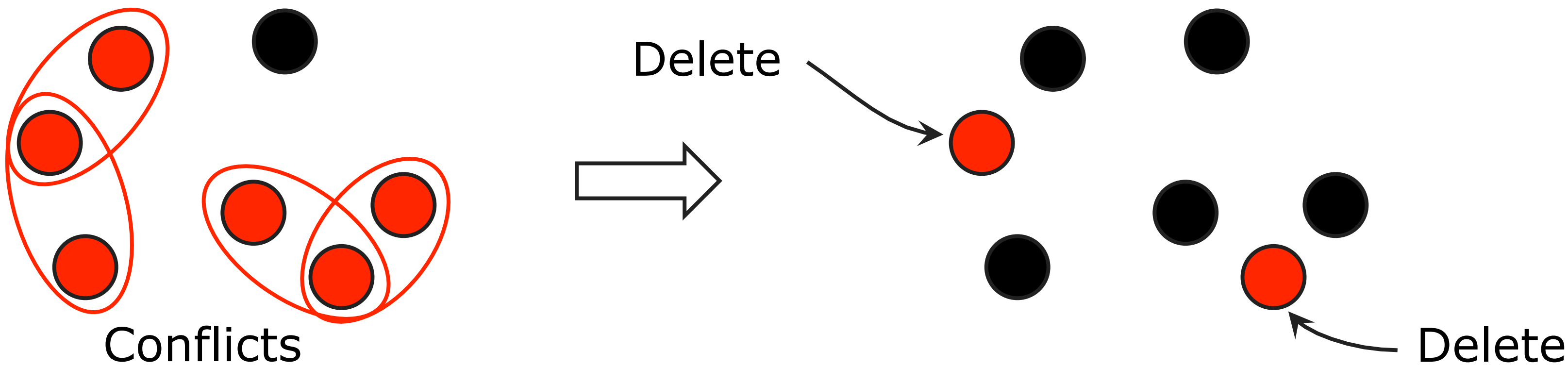
Example: Selecting Airports



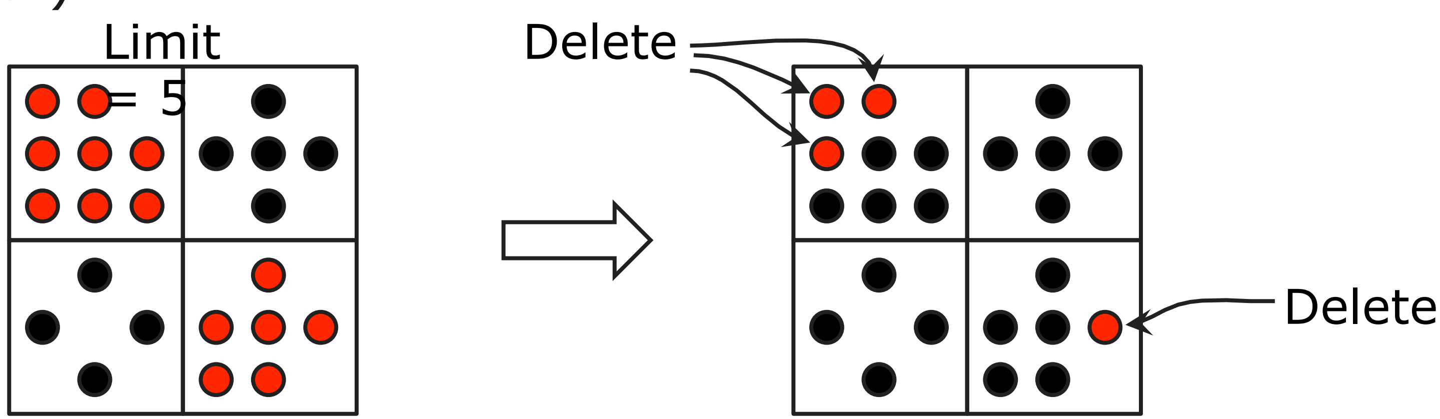
How can we get from “before” to “after”?

Cartographic Constraints

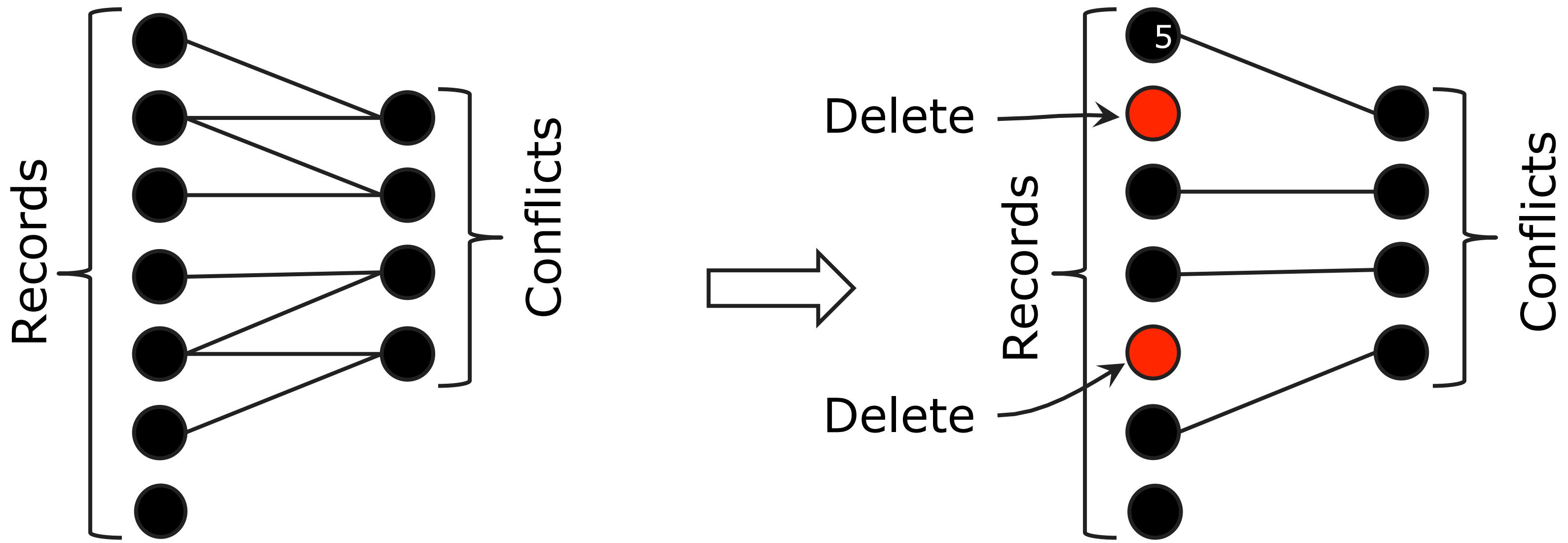
Proximity constraint: minimum distance between records (measured in pixels on screen)



Visibility: maximum records per unit area (within a map "tile")



Optimization Problem



- Cartographic constraints and record importance lead to optimization problem
- Delete minimum weight cover
- Set multicover problem → NP-Hard

Declarative Cartography

```
GENERALIZE
      airports TO airports_zoom
WITH ID airport_id
WITH GEOMETRY wkb_geometry
AT 18 ZOOM LEVELS
WEIGH BY
      num_departures
SUBJECT TO
      proximity 10
      visibility 16
```

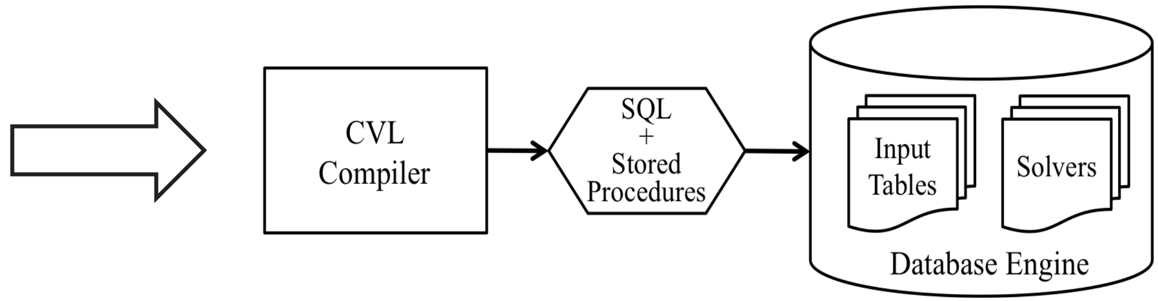
- Creating maps is the job of data-journalists, bloggers, high-level programmers… not mathematicians
- **Cartographic Visualization Language (CVL):** transforms input data into zoomable data
- Constraints expressed in SQL

Compiling CVL into SQL

```
CREATE CONSTRAINT
  proximity AS
```

```
CREATE CONSTRAINT
  visibility AS
```

```
GENERALIZE
  airports TO airports_zoom
WITH ID airport_id
WITH GEOMETRY wkb_geometry
AT 18 ZOOM LEVELS
WEIGH BY
            num_departures
SUBJECT TO
            proximity 10
            visibility 16
```

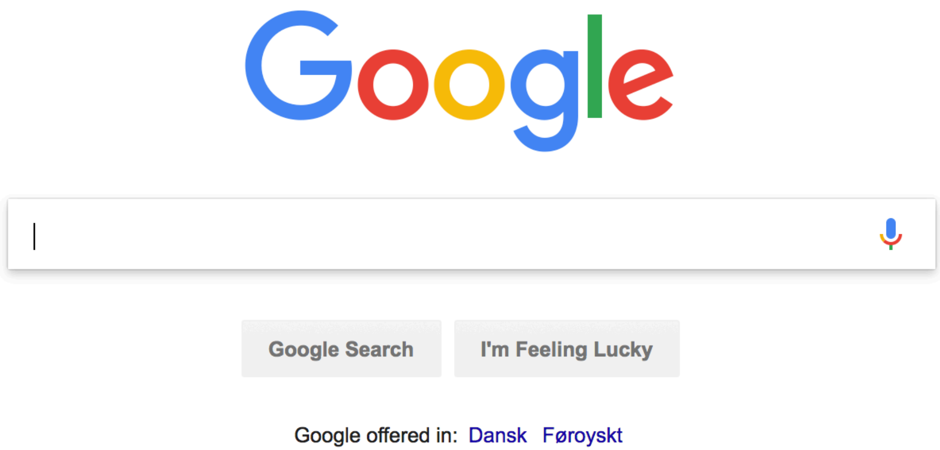


- **Leverage database theory and technology to compute generalization inside DB (in-situ)**
- Our prototype: PostgreSQL + PostGIS + Python + CVXOPT
  - Could be any database, e.g., a parallel one!

Lesson 3: From Software to Services

Did you ever install Google?

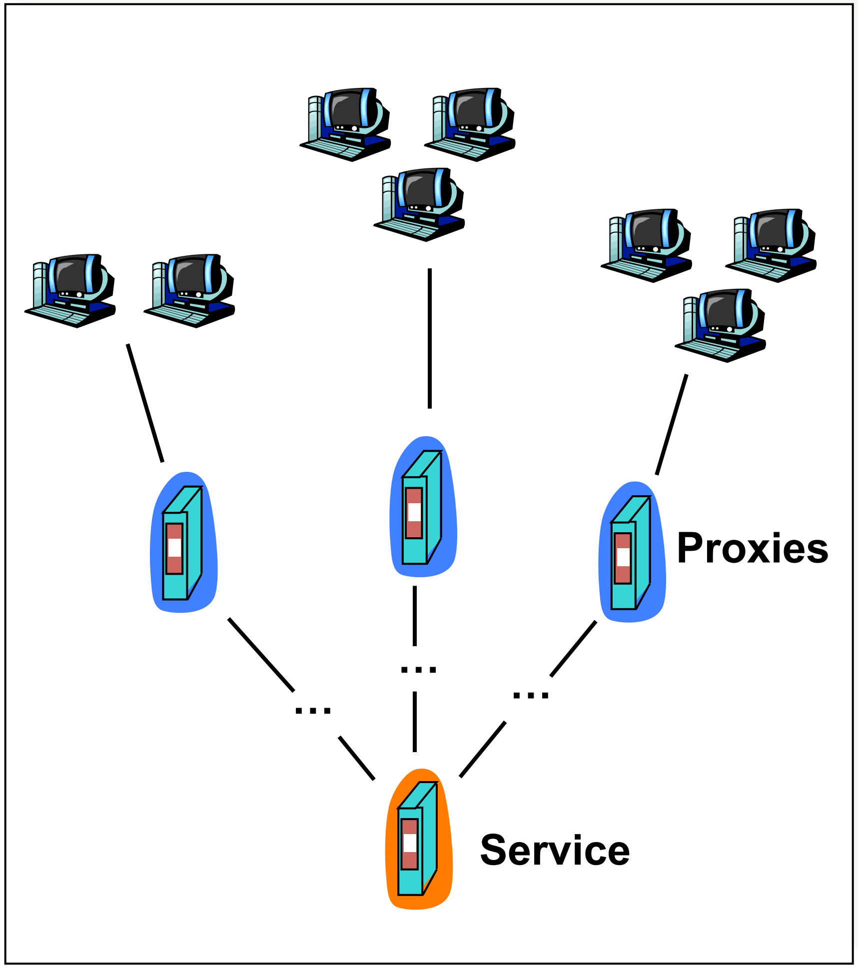
https://www.google.dk/?gws_rd=cr&dcr=0&ei=AxsIWu-8MouE6AS-3YKgDA



- Service hides software and hardware complexity
- Data engineers can make service scalable to **millions or even billions** of users
- Allows for modular interaction
  - Results from search, maps, Q&A, etc

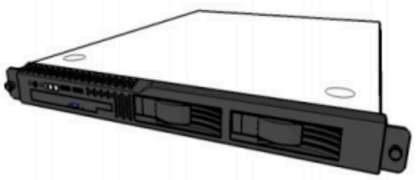
Layering in Web Services

- Services widely exposed on the web, accessible via **HTTP**
- **Proxies** route requests to multiple back-end services and join results
- Services themselves can be implemented on **distributed and parallel architectures**



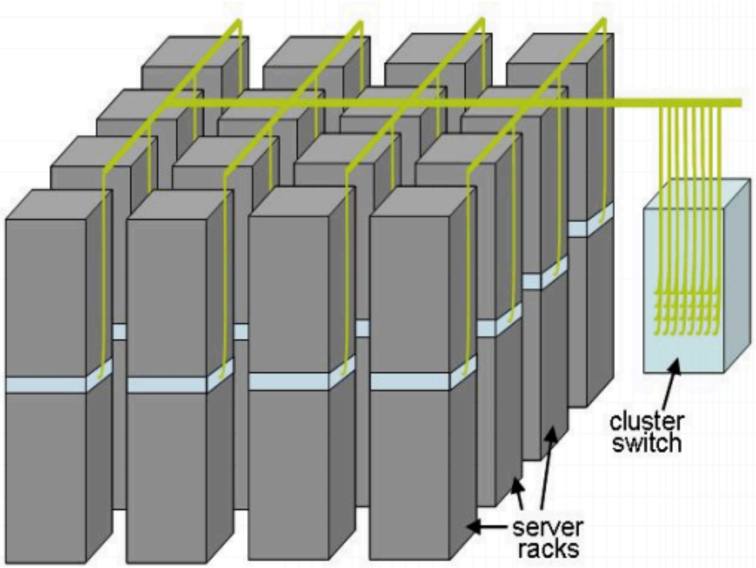
Source: Freedman (partial)

The Machinery



Servers
- CPUs
- DRAM
- Disks

Racks
- 40-80 servers
- Ethernet switch



Clusters

Source: Dean

The Joys of Real Hardware

Typical first year for a new cluster:

- ~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)
- ~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)
- ~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)
- ~1 network rewiring (rolling ~5% of machines down over 2-day span)
- ~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
- ~5 racks go wonky (40-80 machines see 50% packetloss)
- ~8 network maintenances (4 might cause ~30-minute random connectivity losses)
- ~12 router reloads (takes out DNS and external vips for a couple minutes)
- ~3 router failures (have to immediately pull traffic for an hour)
- ~dozens of minor 30-second blips for dns
- ~1000 individual machine failures
- ~thousands of hard drive failures
- slow disks, bad memory, misconfigured machines, flaky machines, etc.

Long distance links: wild dogs, sharks, dead horses, drunken hunters, etc.

Source: Dean

Reliability & Availability

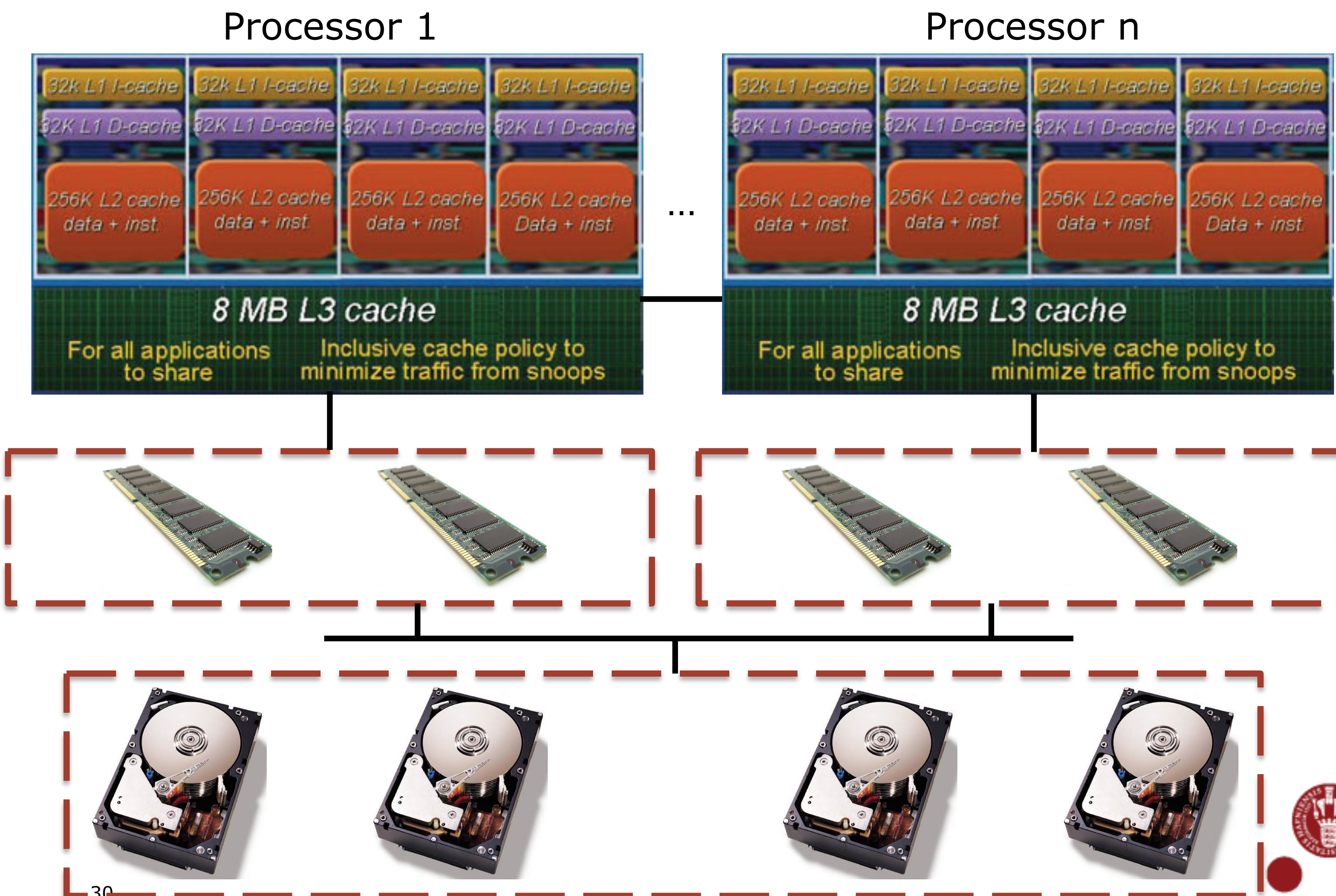
- Things will crash. Deal with it!
  - Assume you could start with super reliable servers (MTBF of 30 years)
  - Build computing system with 10 thousand of those
  - **Watch one fail per day**

- Fault-tolerant software is inevitable

- Typical yearly flakiness metrics
  - 1-5% of your disk drives will die
  - Servers will crash at least twice (2-4% failure rate)

Source: Dean

Complexity lives even inside a single server…



But the picture is not to scale!

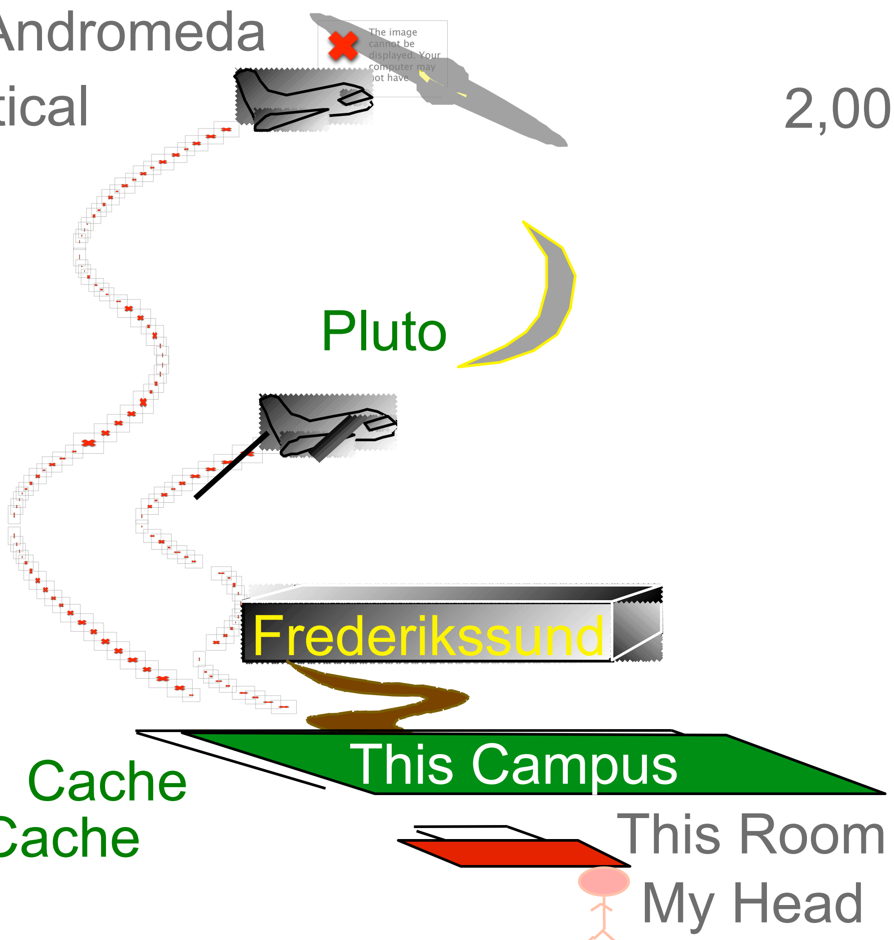
Size of
last-level cache

Size of
memory

What about the size of disk?

Storage Hierarchy

10^9	Tape /Optical Robot	2,000 Years
10^6	Disk	2 Years
100	Memory	1.5 hr
10	On Board Cache	10 min
2	On Chip Cache	
1	Registers	1 min



Source: Gray (partial)

Common Issues in Designing Services

- **Consistency**
  - How to deal with *updates* from multiple clients?
- **Coherence**
  - How to refresh *caches* while respecting consistency?
- **Scalability**
  - What happens to resource usage if we increase the #clients or the #operations?
- **Fault Tolerance**
  - Under what circumstances will the service be unavailable?

Research Highlight: Data Platform for Future Cropping Project

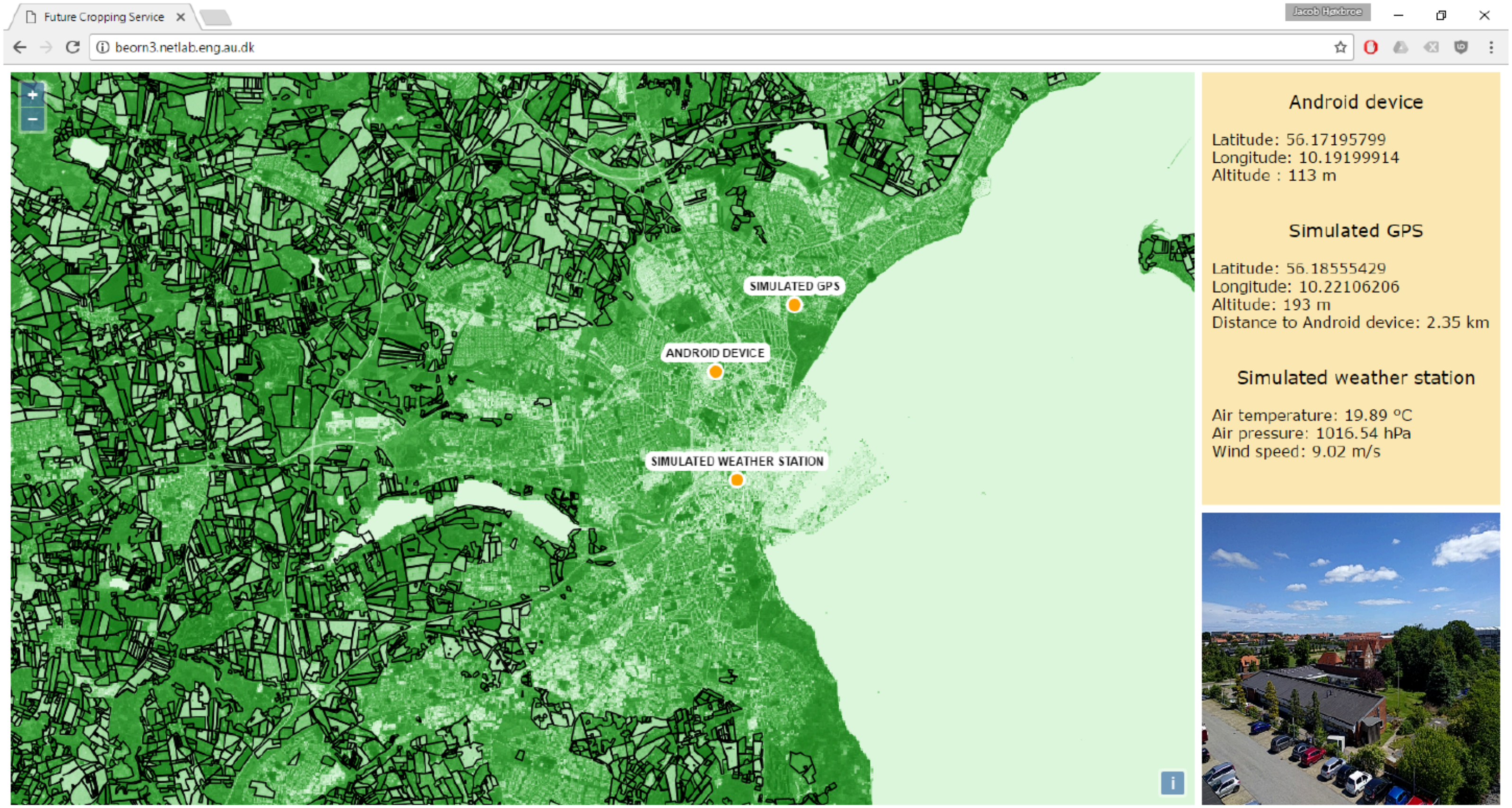
https://en.wikipedia.org/wiki/Spatial_analysis#/media/File:Snow-cholera-map.jpg

Spatial Analysis

Work in collaboration with KU PLEN, Aarhus U, and other partners in Future Cropping project; MSc thesis of Mads Engesgaard Jacobsen and ongoing PhD of Yiwen Wang

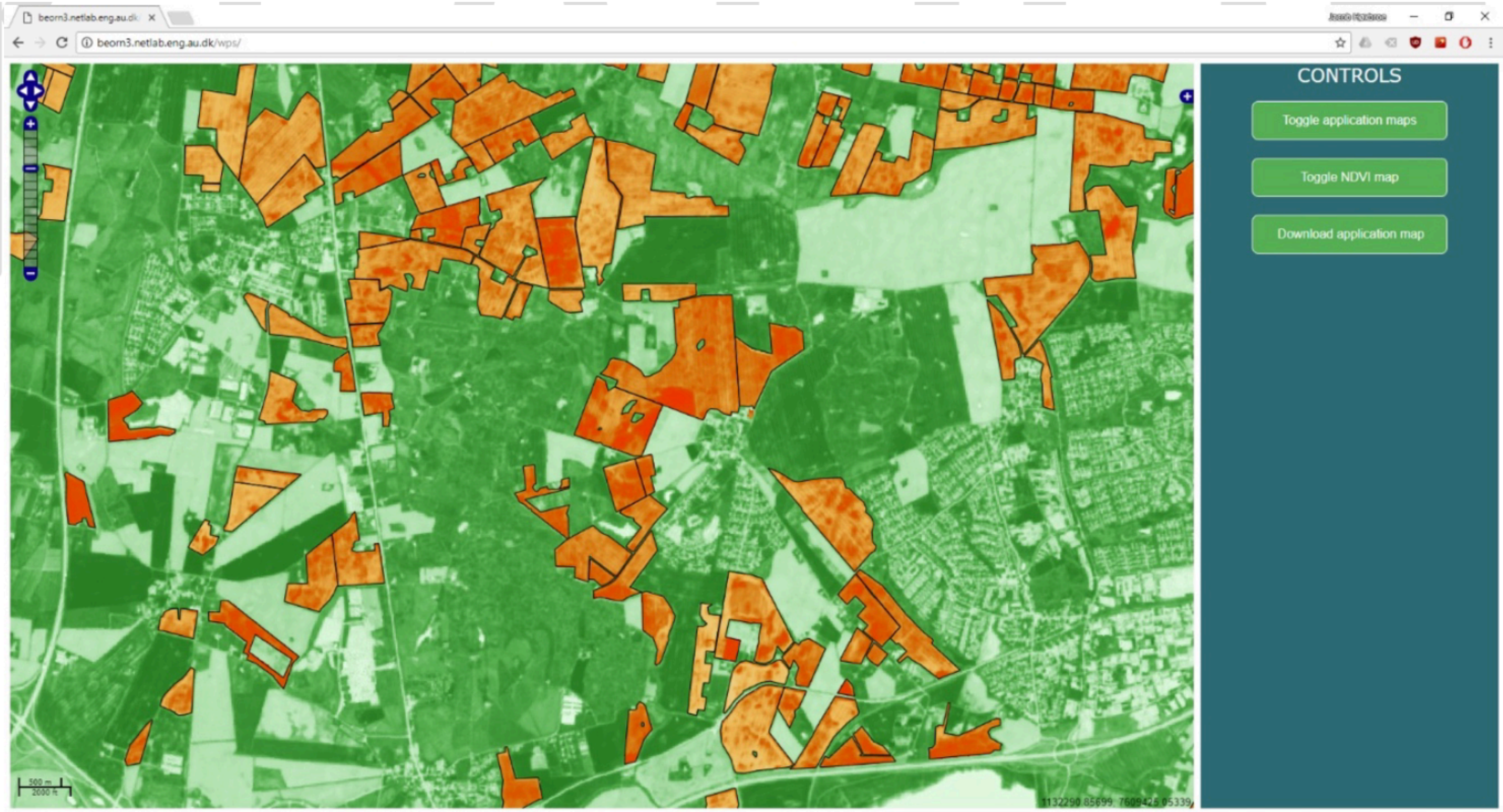
- **Traditional Approach to Data Management in Agriculture**
  - Build BIG database, e.g., data warehouse
  - Lots of time spent mapping schemas, defining what queries to answer
  - Inflexible, high cost, limited to specific questions
- **Future Cropping Data Platform**
  - Build a service-centric data platform
  - Data platform manages and serves geospatial data for analysis services
  - Flexible, pay-as-you-go integration of analytic functions
  - Separation of concerns: Expertise in scalability for data platform

Examples of Services



- Online streaming data from moving objects overlaid with Sentinel-2 satellite data and field polygons
  - Demo available at: http://beorn3.netlab.eng.au.dk/

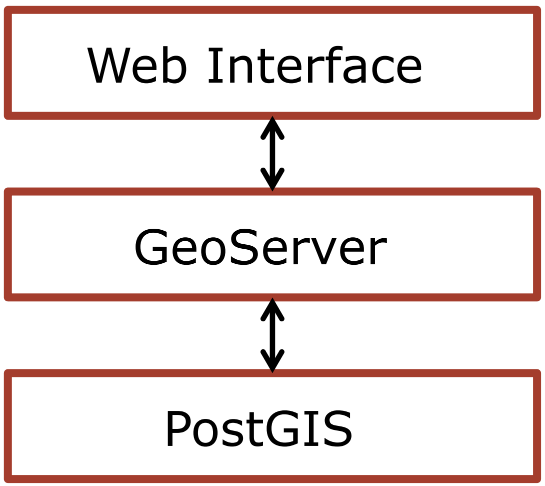
Examples of Services



- Field polygons and NDVI map with option to download fertilizer application maps
  - Demo available at: http://beorn3.netlab.eng.au.dk/wps/

Data Platform Foundations and Trends

- **GeoNode**



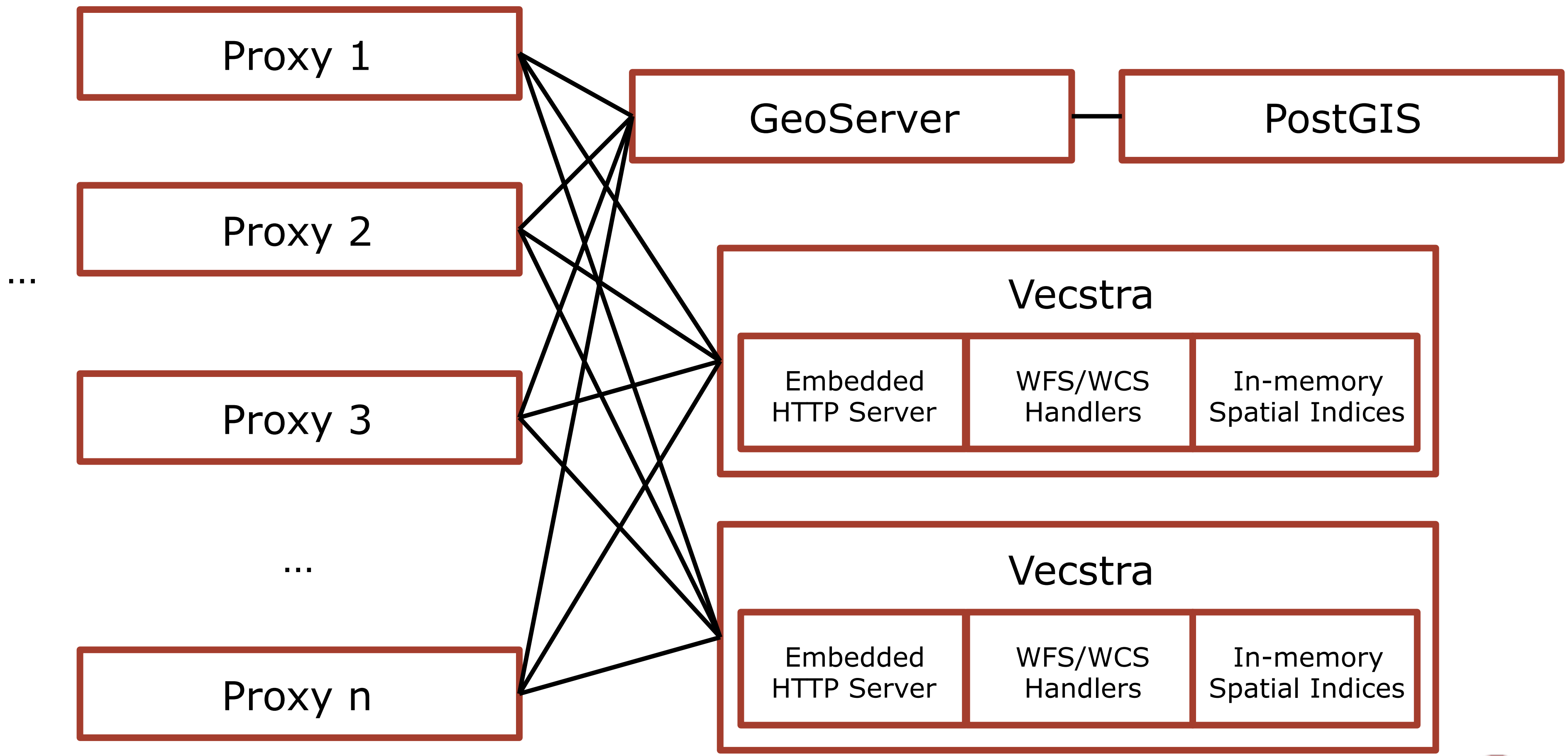
- **Trends**
  - More users
  - More datasets
  - More frequent updates
  - More analytical services

- **Existing systems not enough for the future**
  - General caches, e.g., Redis, Memcached, not specialized for geospatial data
  - Geospatial caches, e.g., GeoWebCache, only cover subset of protocols (WMS)

Vecstra: Core ideas

- **Support for *transparent scalability* on concurrent requests**
  - Communicate with cache through subset of WFS/WCS protocols so as to make solution “drop-in”
  - Reverse proxy layer routes to multiple caches or falls back to GeoNode for advanced functionality
- ***Low latency* in serving data**
  - Employ state-of-the-art in-memory spatial indices
  - Revisit algorithms to speed up specific operations, e.g., geometry intersection, counting queries
- ***Efficiency* in use of computational resources**
  - Design cache multi-threading for performance on multi-core servers

Vecstra Architecture

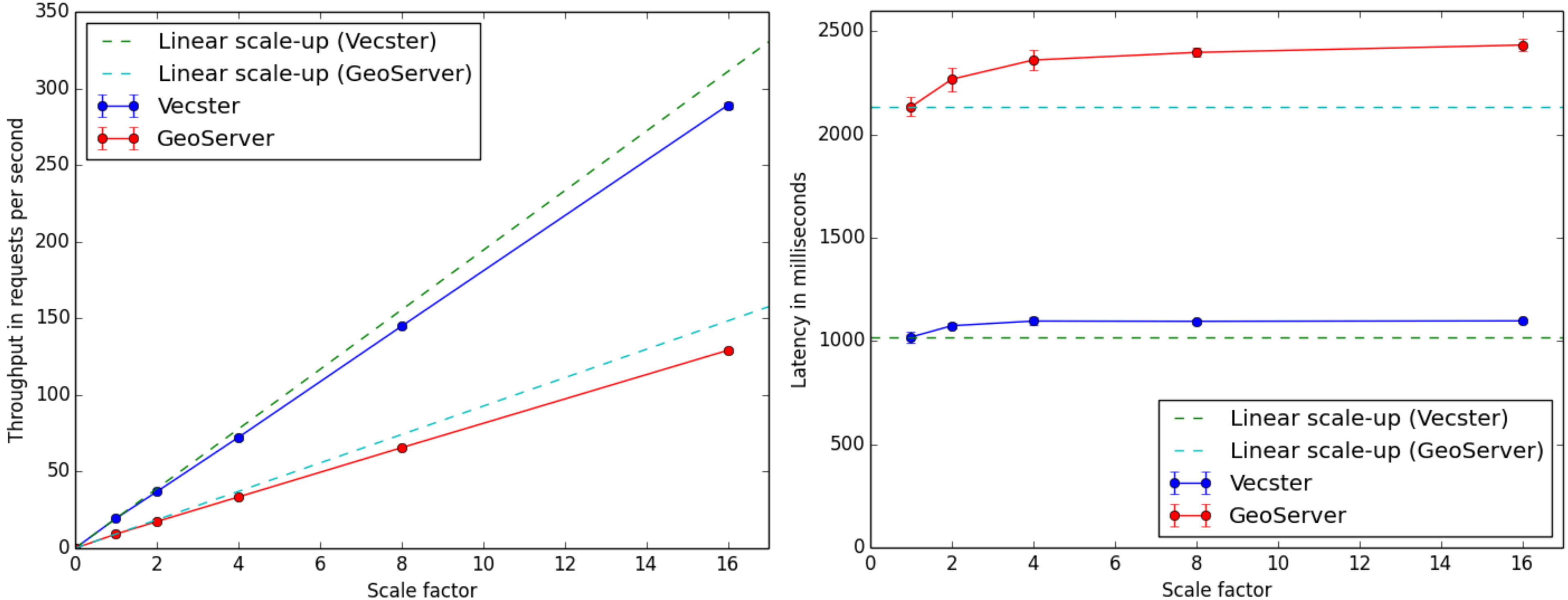


Initial Evaluation: WFS

- **Workload modeling Future Cropping**
  - *Crop Status Service*
    - Field-sized query windows scaled between 0.5x-2x
    - Fields picked uniformly at random, window centered in point within bounding box of field
    - Layers: field polygons, topography, soil, rain distribution (future: also NDVI, climate)
  - No updates for now; need to model update patterns

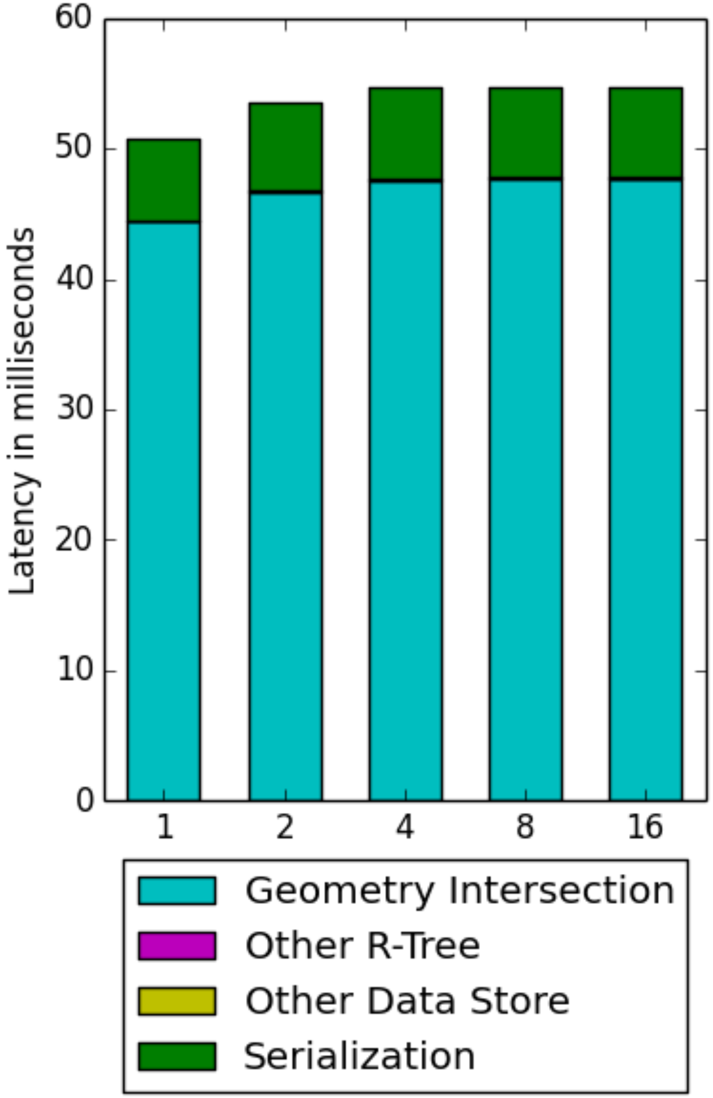
- **Single-node multi-core server**
  - Vector and NDVI layers in data platform as of early 2017 take roughly 23 GB → *in-memory processing*
  - *Server*: *16 cores*; 2 sockets; HT not used; 128 GB RAM
  - Thread affinity or taskset used to limit cores used
  - 20 client threads per server core in separate machine; 10Gbit Ethernet

Initial Evaluation: WFS Spatial Range Query Multi-Core Scalability



- Both systems scale relatively well with increasing numbers of cores in high-load, read-only setup
- Under minimal tuning of both systems, Vecstra shows promise of delivering better latency and scaling efficiency

Initial Evaluation of WFS Spatial Range Query: Where does the time go?



- Server-side latency (HTTP not included)
- Recall we increase client threads (20) to push utilization up
- On the server side, most of the query time is spent in geometry intersection operations, i.e., in refine step of filter-refine scheme
- More work to be done here ☺

Lesson 4: Telemetry Turns Behavior into Data

Usage logs are data!

- **Web log records queries of a web service**
  - User access patterns include **spatial and temporal information**
- **Model user attention and skew in access patterns**
  - For better caching, deployment of computational infrastructure
  - For detection of patterns leading to bias in business decisions
    - Products that users most looked at
    - Breeds that users pay "too much" attention to

Steve Jurvetson - https://www.flickr.com/photos/jurvetson/162116759

Research Highlight: TileHeat

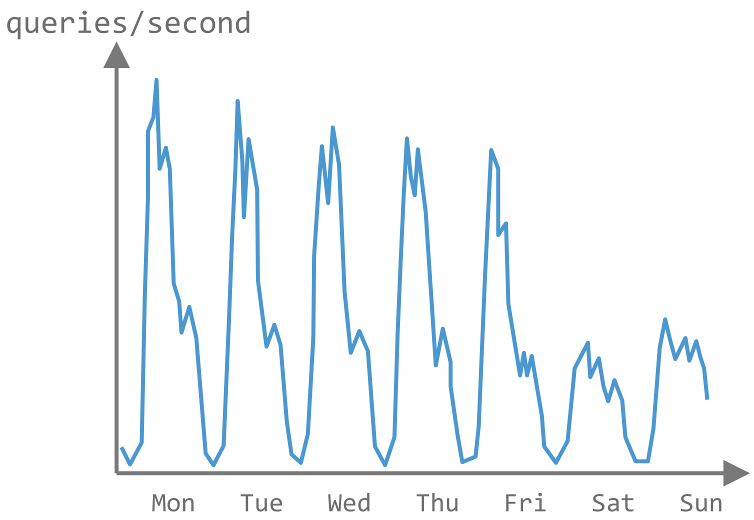
- **Case study of The Digital Map Supply**
  - Most popular WMS web service of Danish National Survey and Cadastre, now Danish Geodata Agency
- **Issues**
  - Render service is slow to compute tiles (for some map services)
  - Bulk data updates by Danish municipalities
  - In combination: Bad performance (for some map services)
- **Data we have analyzed**
  - Request log last 5 years: ~1B requests total
  - Q4 2011 log for most popular map service: ~800K requests per day

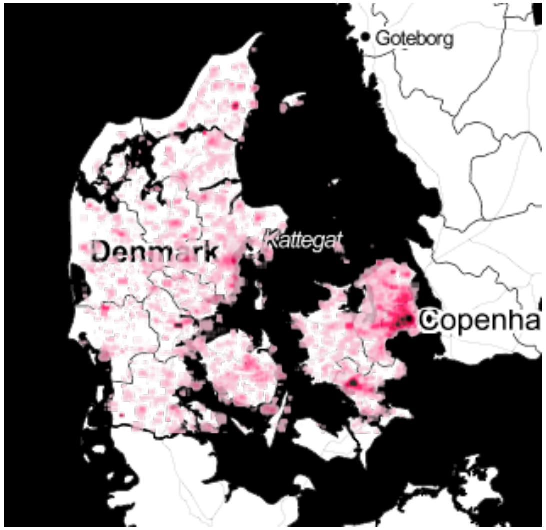
Geodatastyrelsen

http://gst.dk/

Work done in collaboration with P. K. Kefaloukos and M. Zachariasen, results in ACM SIGSPATIAL GIS 2012

Exploitable Properties





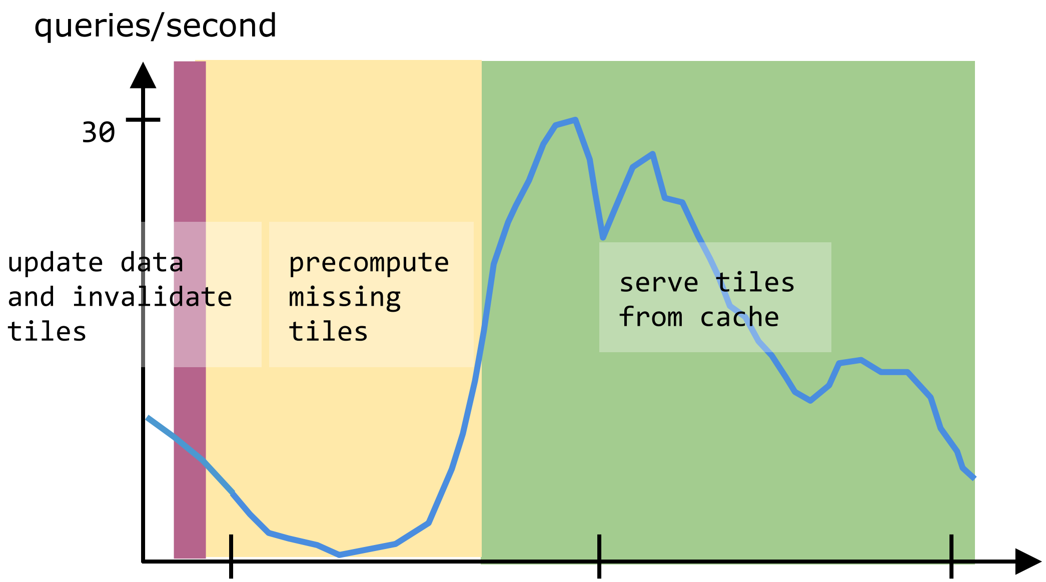
Prediction of where people will look on the map

1) Seasonal variation in load (24-hour, week)
2) We can predict* the tiles people will tend to request
3) Strong skew in requested tiles

*) For the maps we have studied, but not necessarily a priori!

Source: Danish Geodata Agency

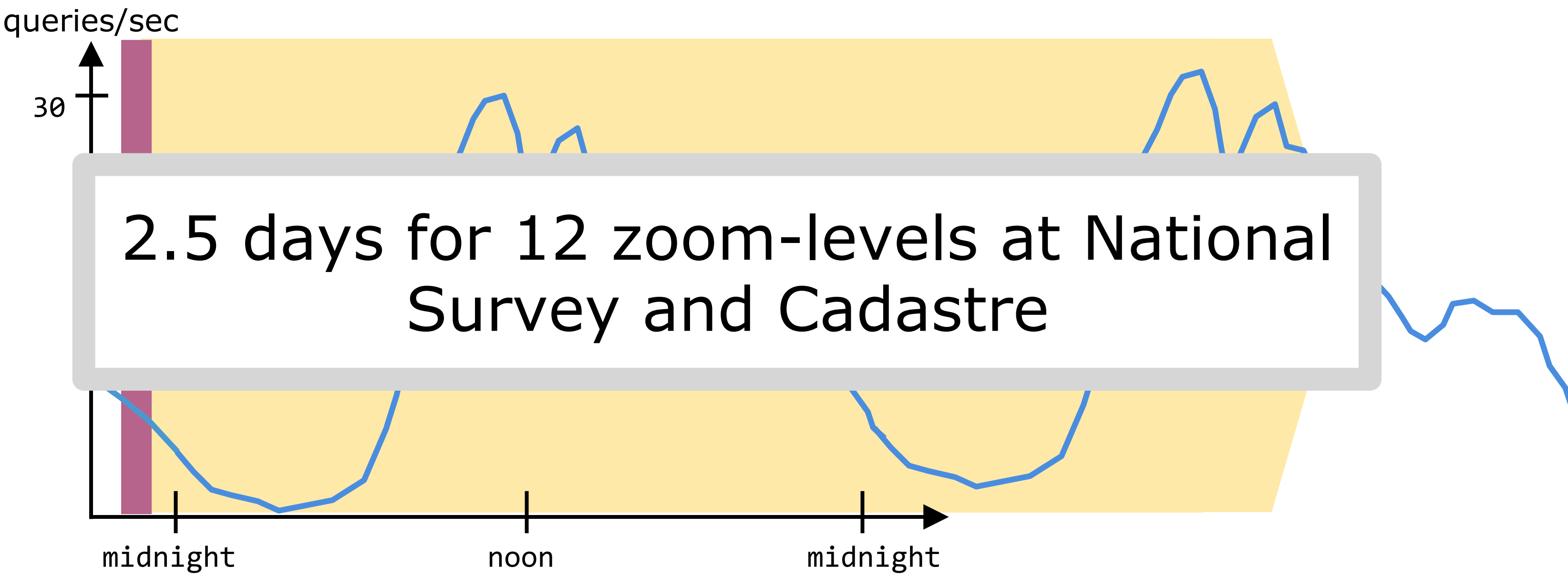
Ideal Situation



- Schedule massive data updates during low load
- Time to refresh cache with new tiles before peak load
- Serve tiles from cache during peak load

Source: Danish Geodata Agency

Problem

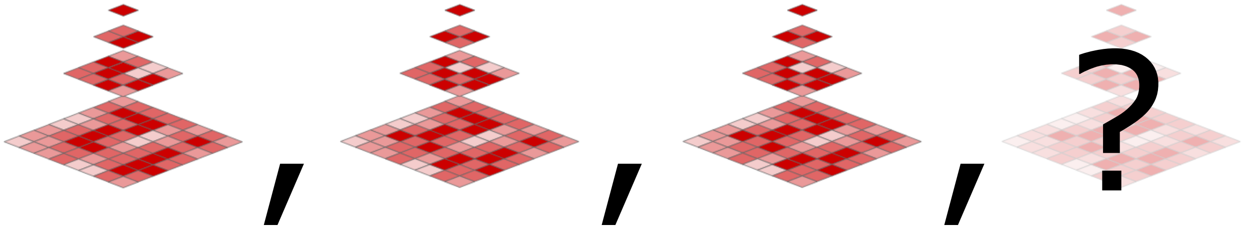


- It takes a long time to generate all tiles
- $O(4^m)$ for m zoom-levels

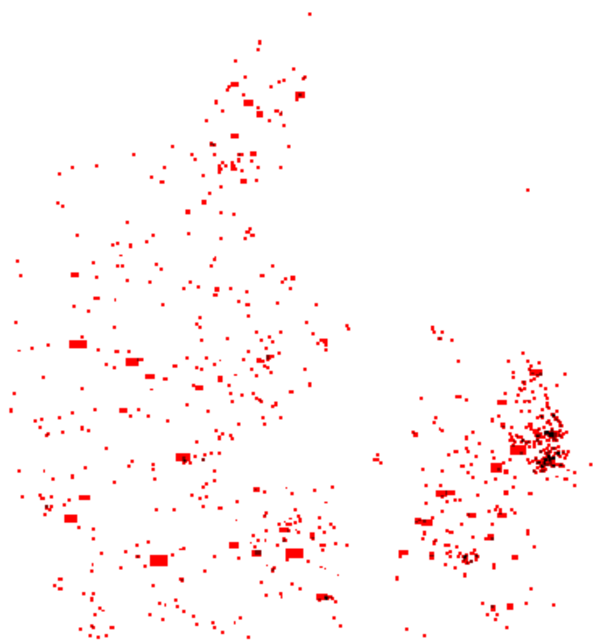
Source: Danish Geodata Agency

Our goal

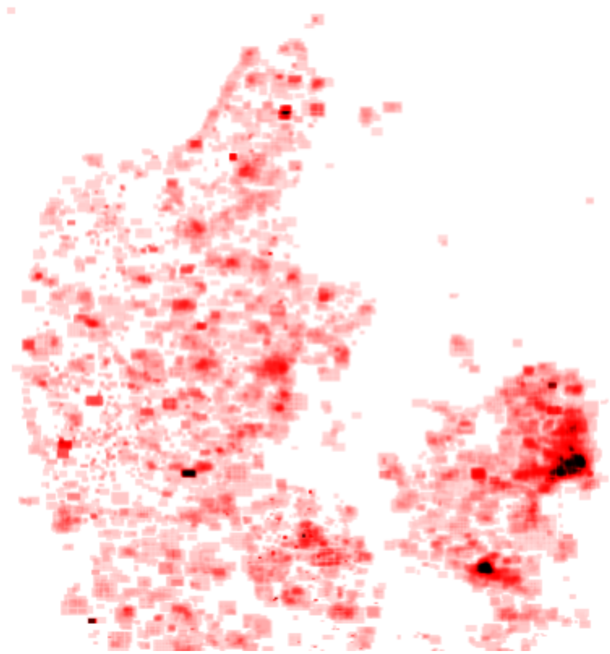
- Predict the heatmap of tomorrow



Heat Dissipation: A Real Example

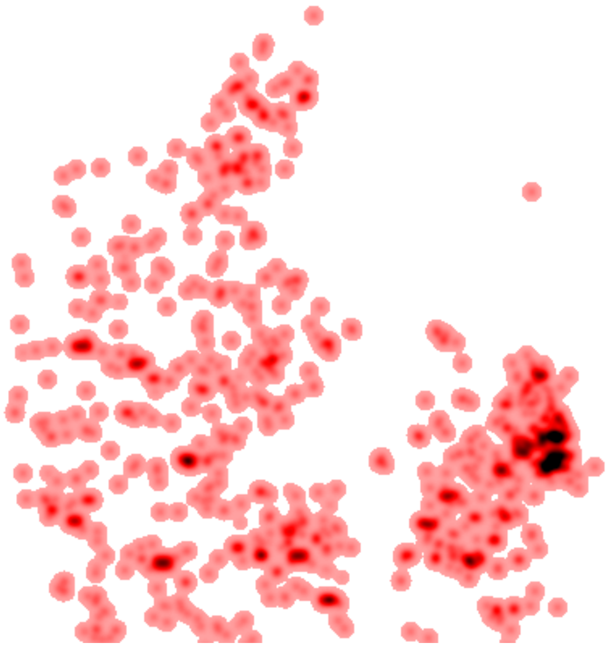


Heatmap based on a small sample of requests (real data)

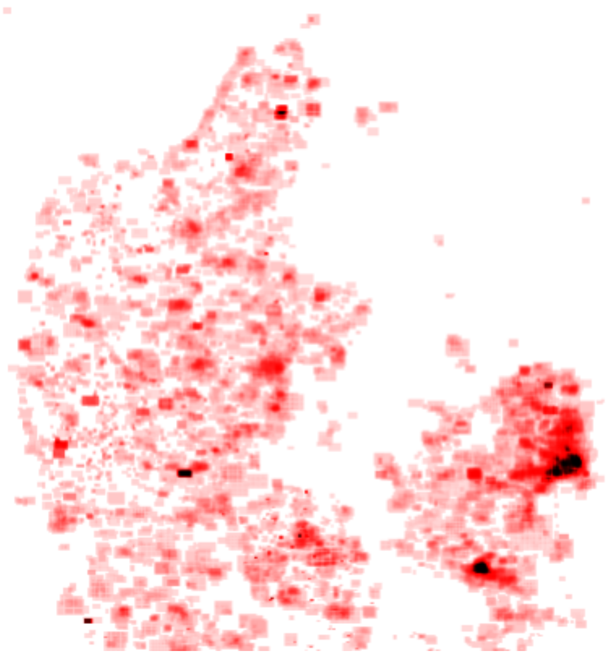


Real heatmap

Heat Dissipation: A Real Example



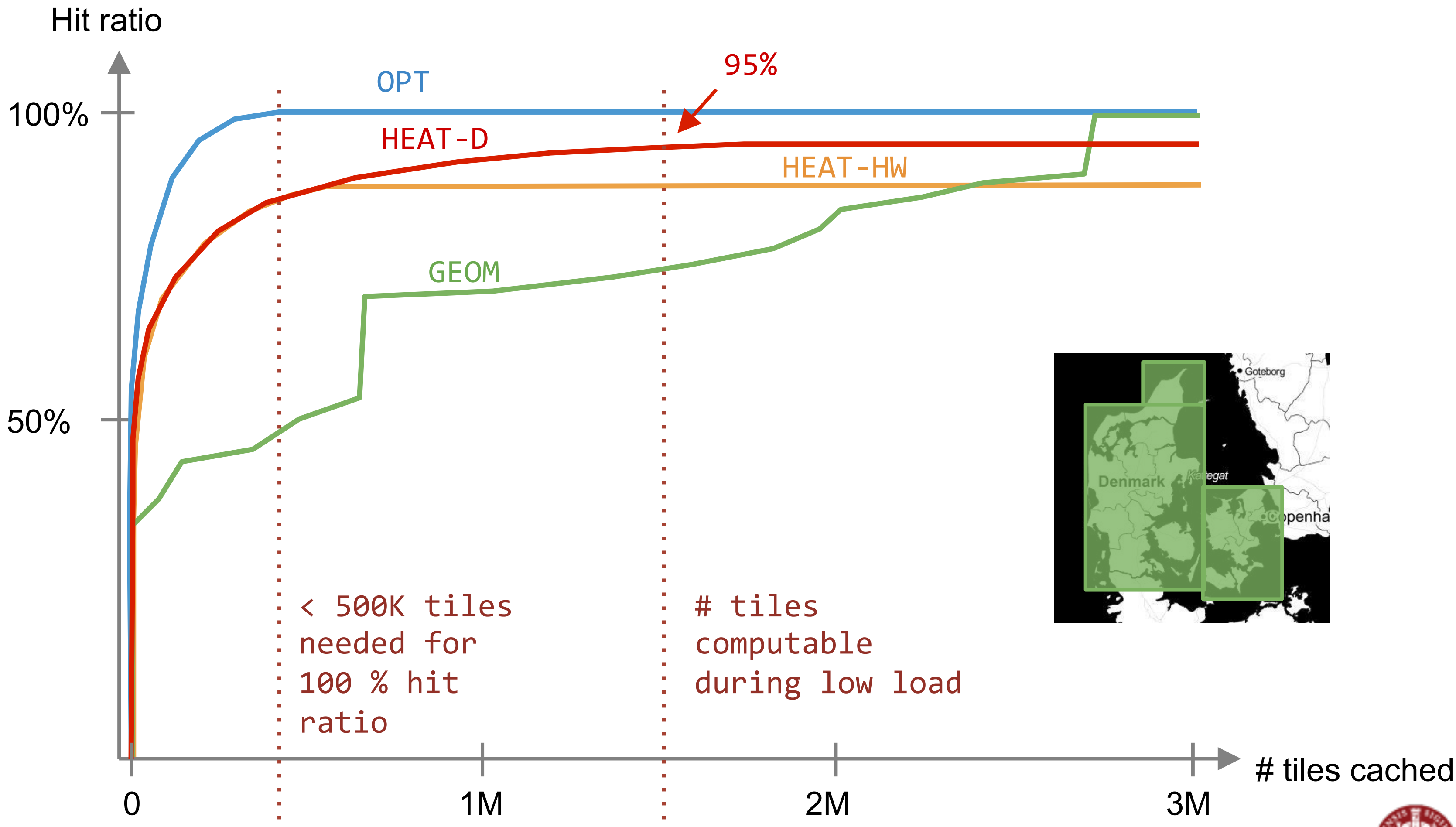
After dissipation



Real heatmap

- In addition to heat dissipation, exponential smoothing model used in predictions to capture variations in time

Evaluation


Hit ratio
100%
50%
OPT
95%
HEAT-D
HEAT-HW
GEOM
< 500K tiles
needed for
100 % hit
ratio
tiles
computable
during low load
Goteborg
Denmark
0
1M
2M
3M
tiles cached

Lesson 5: Embed Intelligence in Services

From querying and calculation to prediction

- **Types and examples of**

)#/media/
_Wykoff.JPG

BEWARE: Data quality is the primary success factor!

obtained so far for all farms
in the country together with
my planned interventions for
this year

Research Highlight: GANDALF project

- **Soil Contamination Management in Urban Areas**
  - Limited sets of pre-selected indicators of potential pollutants in chemical analyses
- **GANDALF**
  - Leverage historical data in spatial interpolation model
  - Enrich existing techniques with machine learning approaches, make data-driven decisions, e.g., for where to sample next
  - Move towards untargeted chemical fingerprinting with high dimensionality and merging with historical data

https://en.wikipedia.org/wiki/Spatial_analysis#/media/File:Snow-cholera-map.jpg

Spatial Analysis

Work in collaboration with KU PLEN, MOE, KMC Nordhavn, and other partners in GANDALF project

Lots of machine learning work going on at DIKU!

Lessons From Managing Geospatial Data

- **Challenge: Big variation in data formats and volume**
  - **Lesson 1: "Cheap" vs. "expensive" data**
  - **Lesson 2: The rise of standardization, open-source software, and large geospatial datasets**
- **Challenge: Large amount of users and potentially complex simultaneous requests**
  - **Lesson 3: From software to services**
  - **Lesson 4: Telemetry turns behavior into data**
- **Challenge: Much labor needed to derive knowledge from varied data**
  - **Lesson 5: Embed intelligence in services**

Conclusion

- **Spatial Applications & Challenges**
- **From Challenges to Lessons**
  - Lesson 1: "Cheap" vs. "expensive" data
  - Lesson 2: The rise of standardization, open-source software, and large geospatial datasets
  - Lesson 3: From software to services
  - Lesson 4: Telemetry turns behavior into data
  - Lesson 5: Embedded intelligence in services
- **Workshop**
  - Groups take lessons as input and discuss how they can be applied to plant phenotyping area
  - Groups summarize discussion work and present in plenum

Thank you!

Background Information

About the Speaker: **Marcos Vaz Salles**

- Associate Professor, University of Copenhagen (**DIKU**)
  - Postdoc: Cornell University
  - PhD: ETH Zurich
- **Expertise:** Database Systems
  - In-memory databases
  - Spatial data
  - Information Integration
  - Cloud Computing
- Co-leader of **Data Management Systems (DMS) Lab**
- **Ongoing Collaborations**
  - Future Cropping consortium: precision agriculture
  - GANDALF consortium: environmental management
  - IDAS: Industrial Data Analysis Service
  - HIPERFIT center: financial apps, risk management