Advanced Database System Architectures

Advanced Topics in Database Management (INFSCI 2711)

Textbook: Database System Concepts - 6th Edition, 2010

Vladimir Zadorozhny, DINS, SCI University of Pittsburgh

1

Database Management System (DBMS)

- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both *convenient* and *efficient* to use
- Database Applications:
  - Banking: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customers, products, purchases

2

Why Use a DBMS?

- Data independence and efficient access.
- Reduced application development time.
- Data integrity and security.
- Uniform data administration.
- Concurrent access, recovery from crashes.
- User-friendly declarative query language.

3

Data Models

- A *data model* is a collection of concepts for describing data.
- The *relational model of data* is the most widely used model today.
  - Main concept: *relation*, basically a table with rows and columns.
  - Every relation has a *schema*, which describes the columns, or fields.

4

Database: Related Tables

customer_id	customer_name	customer_street	customer_city
192-83-7465	Johnson	12 Alma St.	Palo Alto
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account_number	balance
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer_id	account_number
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

5

SQL

- **SQL**: widely used non-procedural database query language
  - Find the name of the customer with customer-id 192-83-7465

    select *customer.customer_name*
    from *customer*
    where *customer.customer_id* = '192-83-7465'

customer_id	customer_name	customer_street	customer_city
192-83-7465	Johnson	12 Alma St.	Palo Alto
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account_number	balance
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer_id	account_number
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

6

Database Architecture

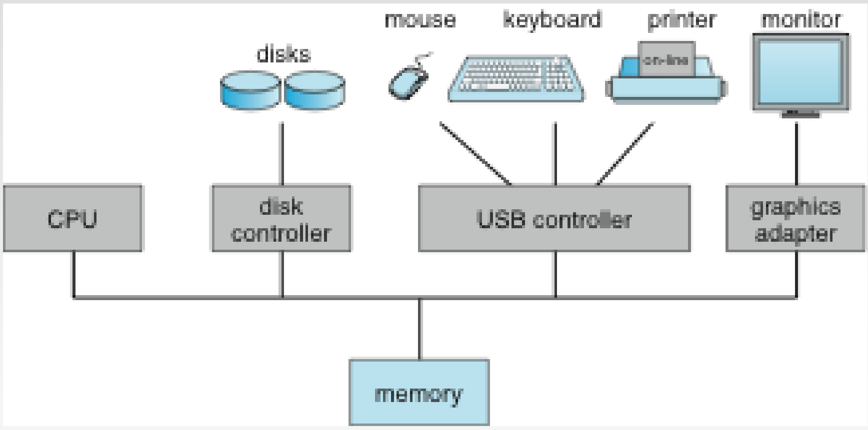
The architecture of a database systems is greatly influenced by the underlying computer system on which the database is running:

- Centralized
- Client-server
- Parallel (multi-processor)
- Distributed

Where we are now: Centralized Systems

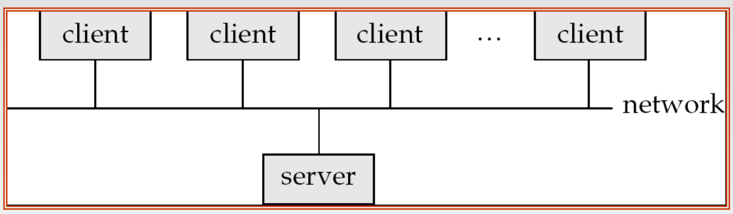
- Run on a single computer system and do not interact with other computer systems.
- General-purpose computer system: one to a few CPUs and a number of device controllers that are connected through a common bus that provides access to shared memory.
- Single-user system (e.g., personal computer or workstation): desk-top unit, single user, usually has only one CPU and one or two hard disks; the OS may support only one user.
- Multi-user system: more disks, more memory, multiple CPUs, and a multi-user OS. Serve a large number of users who are connected to the system vie terminals. Often called *server* systems.

A Centralized Computer System



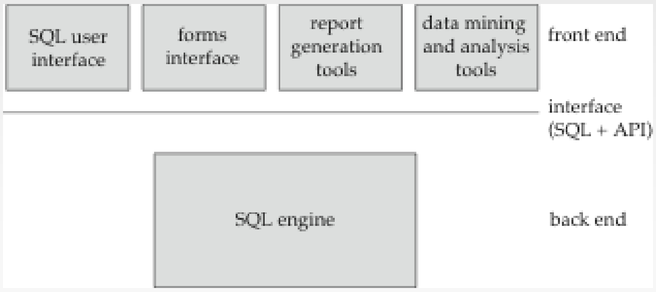
Next: Client-Server Systems

- Server systems satisfy requests generated at *m* client systems:



Client-Server Systems (Cont.)

- Database functionality can be divided into:
  - **Back-end**: manages access structures, query evaluation and optimization, concurrency control and recovery.
  - **Front-end**: consists of tools such as *forms*, *report-writers*, and graphical user interface facilities.
- The interface between the front-end and the back-end is through SQL or through an application program interface.



11

Server System Architecture

- Server systems can be broadly categorized into two kinds:
  - **transaction servers** which are widely used in relational database systems, and
  - **data servers**, used in object-oriented database systems

12

Transaction Servers

- Also called **query server** systems or SQL *server* systems
  - Clients send requests to the server
  - Transactions are executed at the server
  - Results are shipped back to the client.
- *Open Database Connectivity* (ODBC) is a C language application program interface standard from Microsoft for connecting to a server, sending SQL requests, and receiving results.
- JDBC standard is similar to ODBC, for Java

13

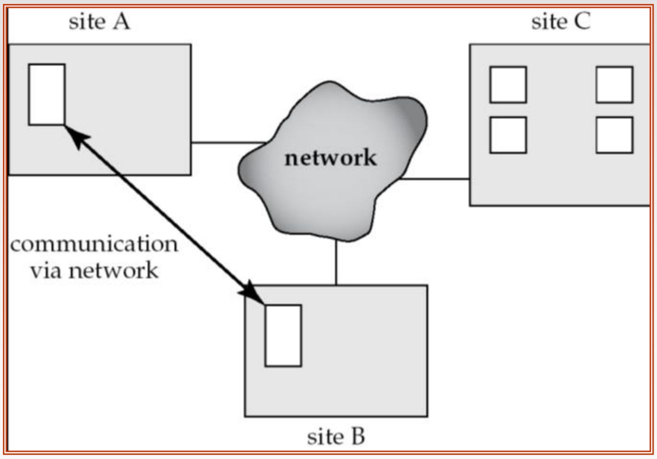
Data Servers

- Data are shipped to clients where processing is performed.
- This architecture requires full back-end functionality at the clients.
- Used in many object-oriented database systems
- Issues:
  - Page-Shipping versus Item-Shipping (tuple, or object)
  - Locking
  - Data Caching

14

Next: Distributed Systems

- Data spread over multiple machines (also referred to as **sites** or **nodes**).
- Network interconnects the machines
- Data shared by users on multiple machines



15

Distributed Databases

- Homogeneous distributed databases
  - Same software/schema on all sites, data may be partitioned among sites
  - Goal: provide a view of a single database, hiding details of distribution
- Heterogeneous distributed databases
  - Different software/schema on different sites
  - Goal: integrate existing databases to provide useful functionality
- Differentiate between *local* and *global* transactions
  - A local transaction accesses data in the *single* site at which the transaction was initiated.
  - A global transaction either accesses data in a site different from the one at which the transaction was initiated or accesses data in several different sites.

16

Trade-offs in Distributed Systems

- Sharing data – users at one site able to access the data residing at some other sites.
- Autonomy – each site is able to retain a degree of control over data stored locally.
- Higher system availability through redundancy — data can be replicated at remote sites, and system can function even if a site fails.
- Disadvantage: added complexity required to ensure proper coordination among sites.
  - Software development cost.
  - Greater potential for bugs.
  - Increased processing overhead.

17

Heterogeneous Distributed Databases

- Different software/schema on different sites
- Goal: integrate existing databases to provide useful functionality

18

Information Integration from a DB Perspective

- Information Integration Challenge
  - *Given*: data sources S_1, ..., S_k (DBMS, web sites, ...) and user questions Q_1,...,Q_n that can be answered using the S_i
  - *Find*: the answers to Q_1, ..., Q_n
- The Database Perspective: source = "database"
  - ⇒ S_i has a schema
  - ⇒ S_i can be queried
  - ⇒ define virtual (or materialized) integrated views V over S_1,...,S_k using database query languages
  - ⇒ questions become queries Q_i against V(S_1,...,S_k)

19

Querying Web Data from a DB Perspective

- Manual navigation over multilevel links: inefficient

  Find the top selling book on C++ at Amazon ?

- Objective: database-like declarative queries:

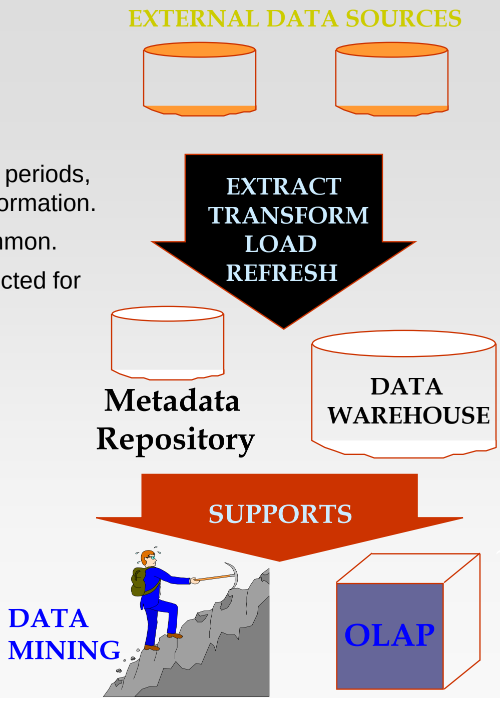
```
select bookTitle
from   Amazon
where bookTopic = "C++" and
          bookSalesRank > all  ( select  bookSalesRank
                                  from Amazon
                                  where bookTopic = "C++" )
```

Handling semi-structured and unstructured data?

20

Data Warehousing

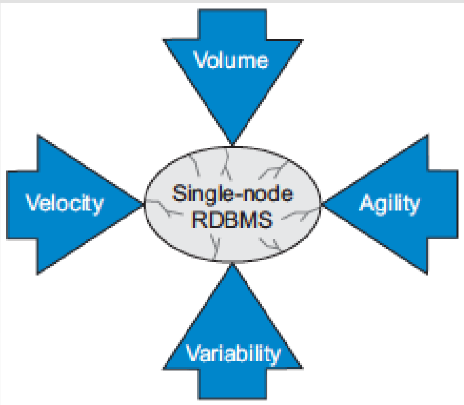
- Integrated data spanning long time periods, often augmented with summary information.
- Several gigabytes to terabytes common.
- Interactive response times expected for complex queries; ad-hoc updates uncommon.



21

NoSQL Business Drivers

- Many organizations supporting single-CPU relational systems have come to a crossroads: the needs of their organizations are changing. Businesses have found value in rapidly capturing and analyzing large amounts of variable data, and making immediate changes in their businesses based on the information they receive.



22

Types of NoSQL data stores

Type	Typical usage	Examples
Key-value store—A simple data storage system that uses a key to access a value	• Image stores • Key-based filesystems • Object cache • Systems designed to scale	• Berkeley DB • Memcache • Redis • Riak • DynamoDB
Column family store—A sparse matrix system that uses a row and a column as keys	• Web crawler results • Big data problems that can relax consistency rules	• Apache HBase • Apache Cassandra • Hypertable • Apache Accumulo
Graph store—For relationship-intensive problems	• Social networks • Fraud detection • Relationship-heavy data	• Neo4j • AllegroGraph • Bigdata (RDF data store) • InfiniteGraph (Objectivity)
Document store—Storing hierarchical data structures directly in the database	• High-variability data • Document search • Integration hubs • Web content management • Publishing	• MongoDB (10Gen) • CouchDB • Couchbase • MarkLogic • eXist-db • Berkeley DB XML

23

Challenge of Unstructured Data: Database Management vs Information Retrieval

- Data: DB: Set of Tables with well defined schema
  IR: Set of (text) documents
- Goal: DB: Find an accurate response to a user query
  IR: Retrieve documents with information that is relevant to user's information need

24

24

Querying unstructured data

- Which plays of Shakespeare contain the words ***Brutus*** *AND* ***Caesar*** but *NOT* ***Calpurnia***?
  - One could grep all of Shakespeare's plays for ***Brutus*** and ***Caesar,*** then strip out lines containing ***Calpurnia***?
    - Slow (for large corpora)
    - *NOT* ***Calpurnia*** is non-trivial
    - Other operations (e.g., find the word ***Romans*** near ***countrymen***) not feasible
    - Ranked retrieval (best documents to return)

25

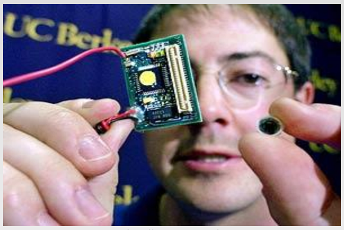
25

What Next?

More challenging network environments …

26

Wireless Sensors



Courtesy: http://www.economist.com

- Small wireless devices (motes)
- Low cost, battery powered
- Sense physical phenomena
  - Light, temperature, vibration, acceleration, AC power, humidity.
- Process/aggregate data
- Communicate

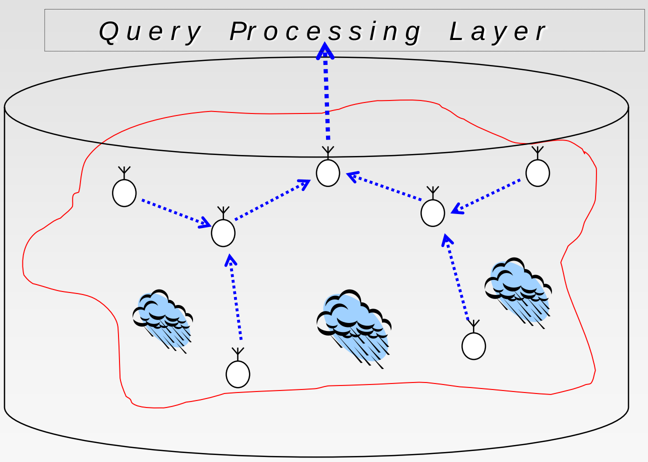
Applications of Wireless Sensor Networks:

Information tracking systems (e.g., airport security);
Children monitoring in metro areas;
Product transition in warehouse networks;
Fine-grained weather measurements;
Structural Health Monitoring

27

Sensor Databases

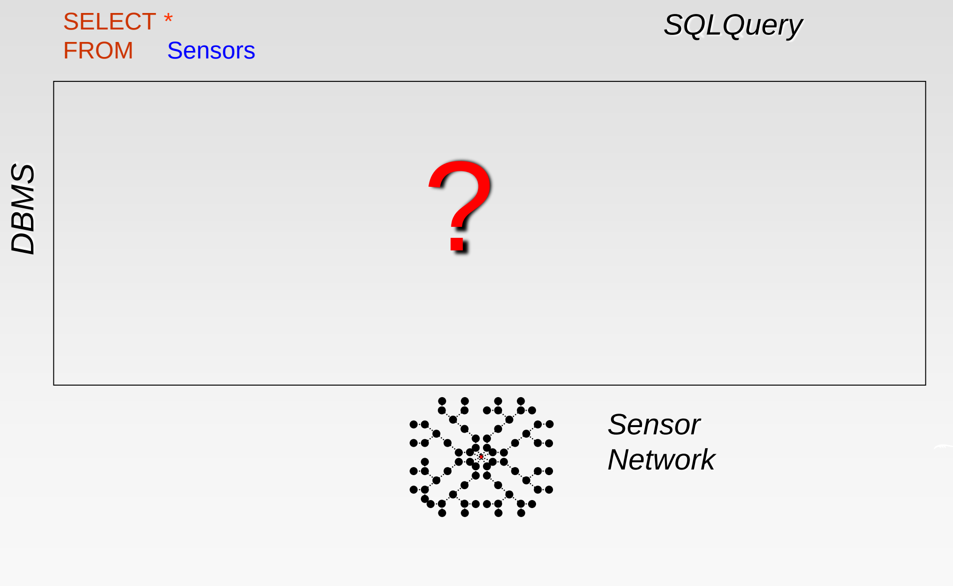
SELECT avg(*rainFallLevel*)
FROM Sensors;



Network is a Database !

28

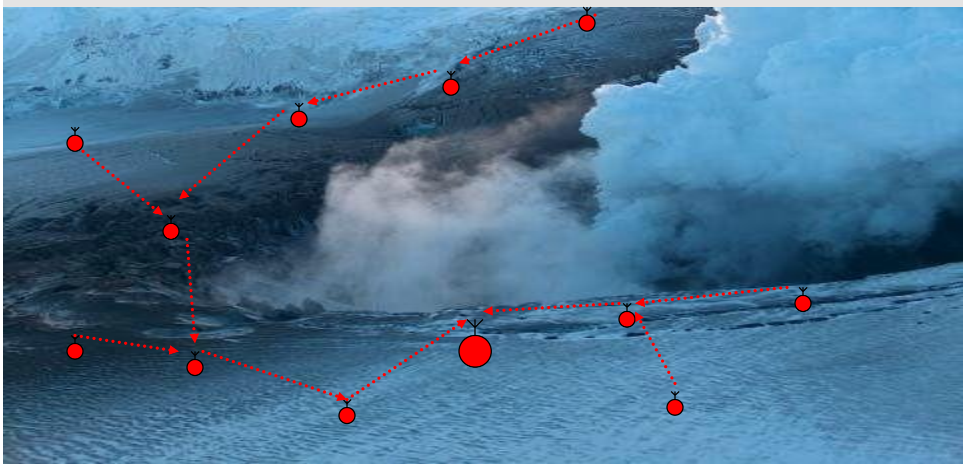
Sensor Database Query Processing ?



29

Mobility: Cool Applications

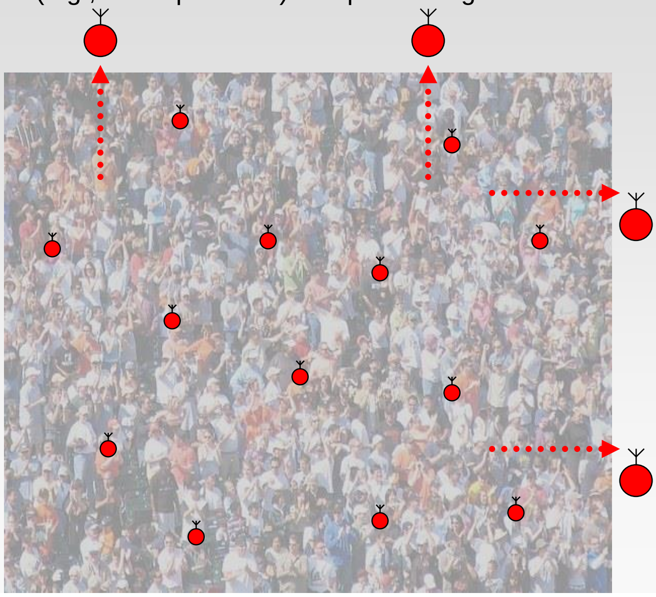
• E.g., a team of cooperative mobile robots can be considered as a wireless sensornet.
• Deployed in conjunction with stationary sensor nodes
• Acquire and process data for surveillance, tracking, environmental monitoring, or execute search and rescue operations.



30

Application #2

• Large-scale human health monitoring with body sensors reporting critical health parameters (e.g., blood pressure) to a processing station.

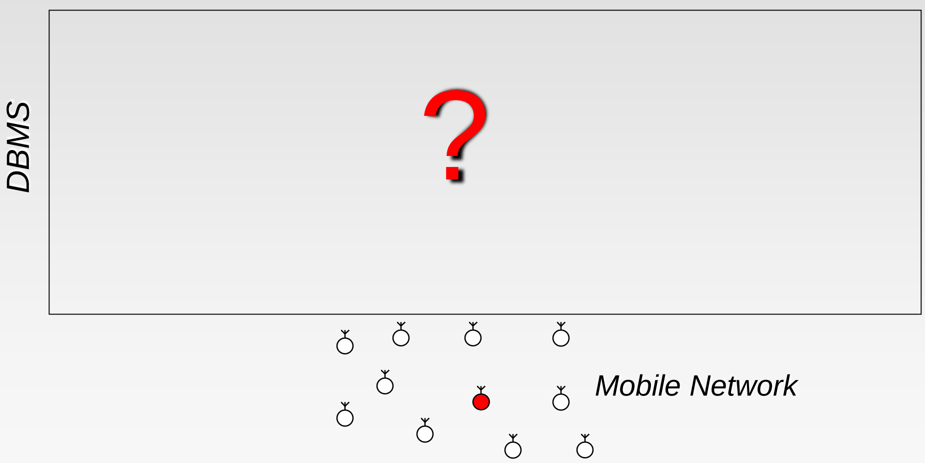


31

Mobile Database Query Processing?

SELECT *Environmental_Conditions*
FROM Sensors

SQLQuery



32

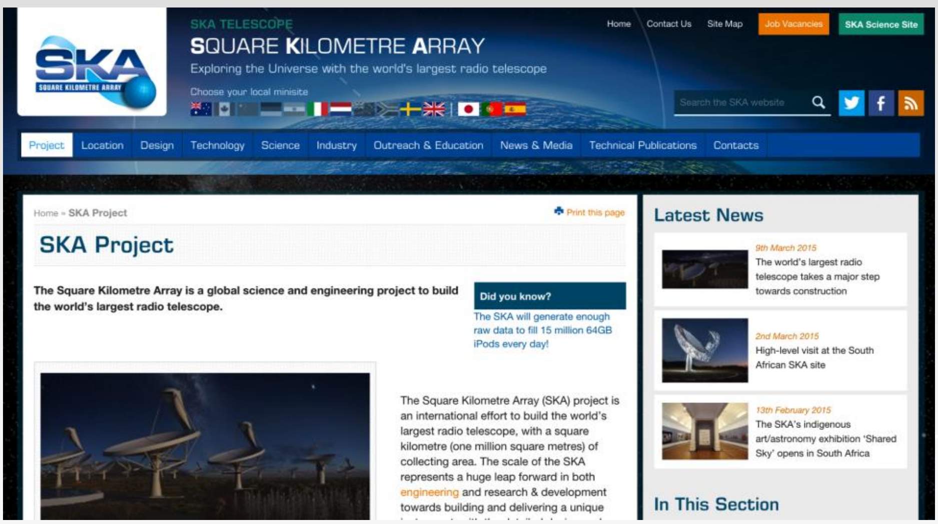
What Next?

Big Data Challenge

33

Big Research Data: Square Kilometre Array

https://www.skatelescope.org/



34

34

Big Research Data: Square Kilometre Array

https://www.skatelescope.org/

The total collecting area of the SKA will be one square kilometers, or 1,000,000 square meters. This will make the SKA the largest radio telescope array ever constructed, by some margin.

To achieve this, the SKA will use several thousand dish (high frequency) and many more low frequency and mid-frequency aperture array telescopes, with the several thousand dishes each being 15 metres in diameter.

The SKA will be so sensitive that it will be able to detect an airport radar on a planet 50 light years away.

The data collected by the SKA in a single day would take nearly two million years to playback on an ipod.

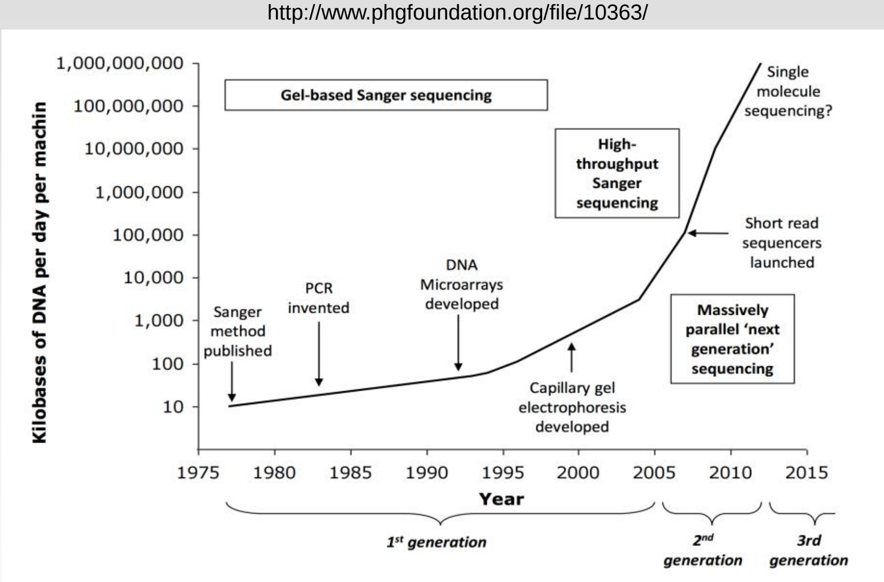
The dishes of the SKA will produce **10 times the global internet traffic**.

35

35

Big Research Data: Evolution of DNA sequencing technology

http://www.phgfoundation.org/file/10363/



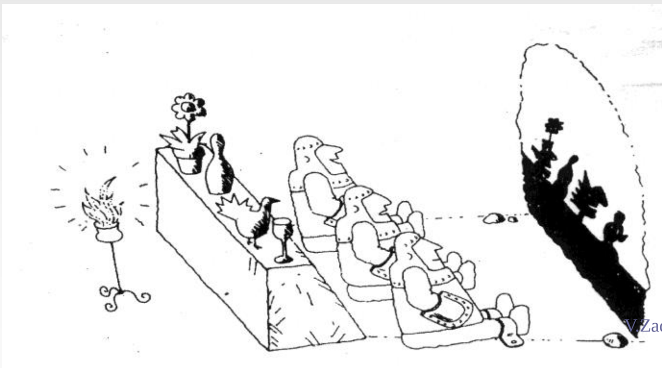
36

Problem of Data Fusion

• Data fusion is a process of resolving data conflicts due to redundancy and inconsistency in data extracted from multiple data sources.

• Domains: multi-sensor data fusion, human-centered information fusion methods, information fusion for data integration.

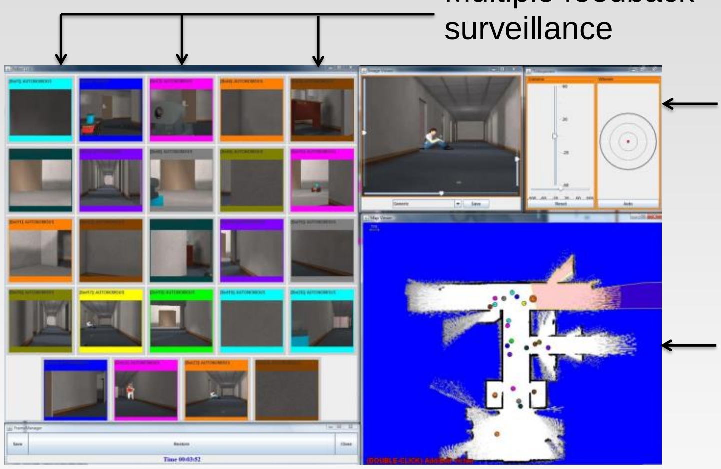
Plato Cave Metaphor:



V.Zadorozhny 37

Multiple feeds problem in Multi-robot Search and Rescue

Multiple feedback surveillance

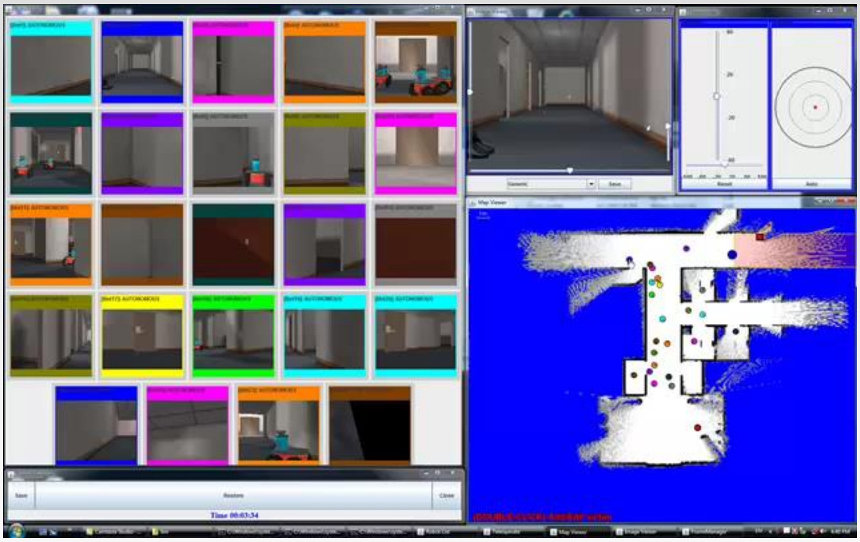


Teleoperation of single robot

Multi-robot control

38

Multiple feeds problem in Multi-robot Search and Rescue



39

39

What Next?

Back to schedule …

40