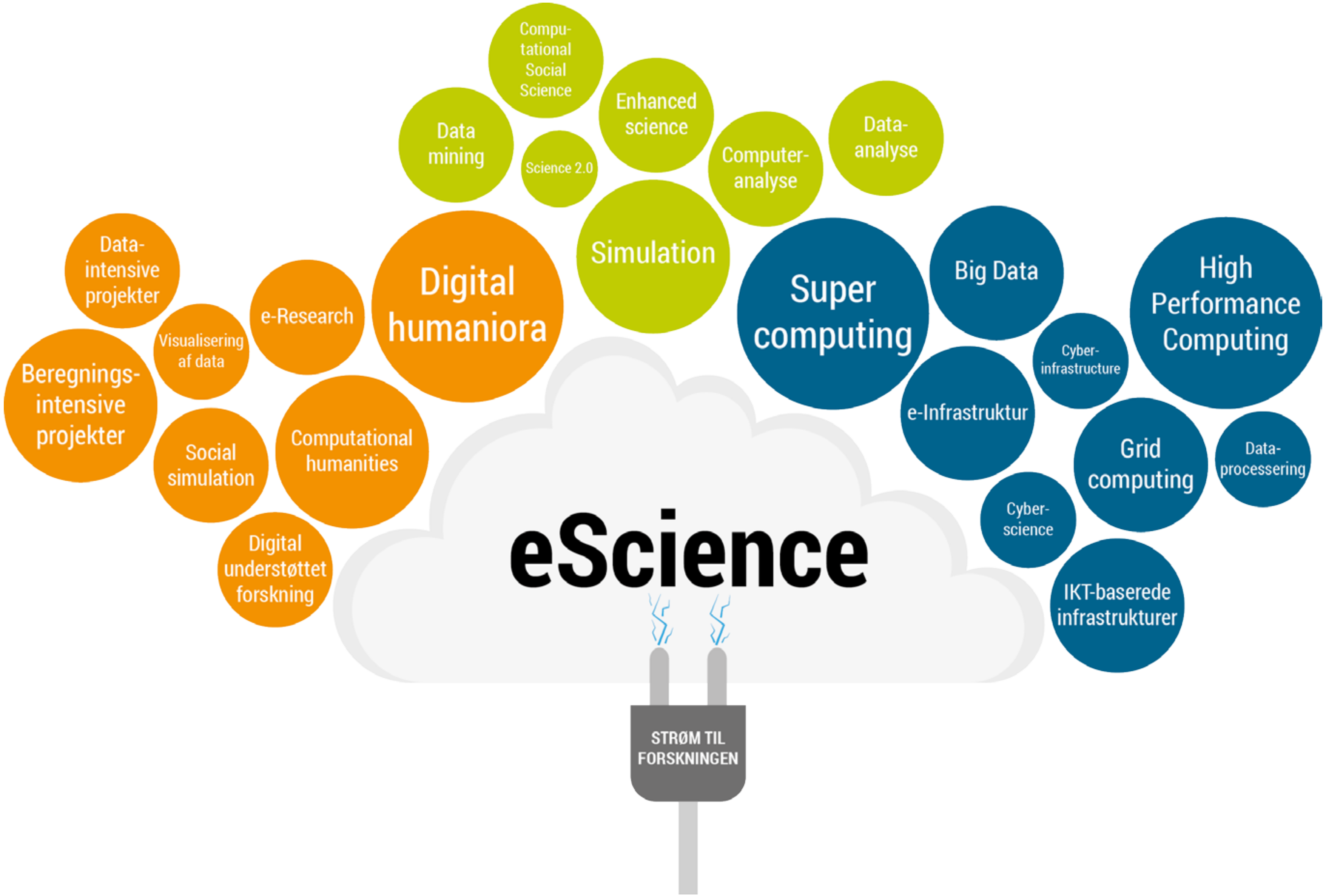
Big data as the future of information technology

Alexander BOGDANOV, Alexander DEGTYAREV, Vladimir KORKHOV

Thurein KYAW LWIN, St-Petersburg State University, RUSSIA

Plan of presentation

1. **Boost of DATA and need of definitions**
2. **Naïve definition and its consequences**
3. **DATA API**
4. **BIG DATA Ecosystem**
5. **Necessary experiments**



Computational Social Science

Data mining

Science 2.0

Enhanced science

Computer-analyse

Data-analyse

Simulation

Data-intensive projekter

e-Research

Digital humaniora

Visualisering af data

Beregnings-intensive projekter

Social simulation

Computational humanities

Digital understøttet forskning

Super computing

Big Data

High Performance Computing

Cyber-infrastructure

e-Infrastruktur

Grid computing

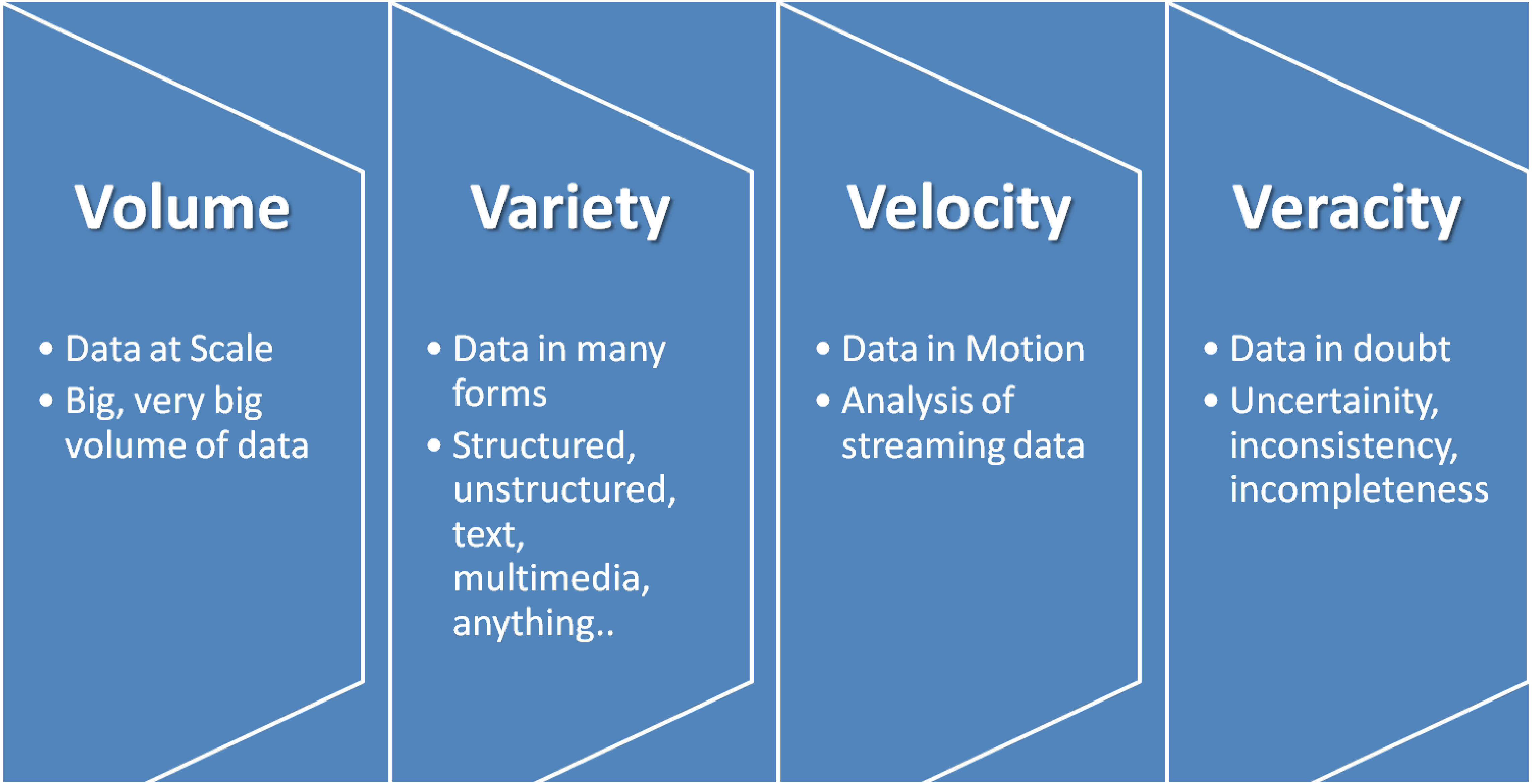
Data-processering

Cyber-science

IKT-baserede infrastrukturer

eScience

STRØM TIL FORSKNINGEN



Volume

- Data at Scale
- Big, very big volume of data

Variety

- Data in many forms
- Structured, unstructured, text, multimedia, anything..

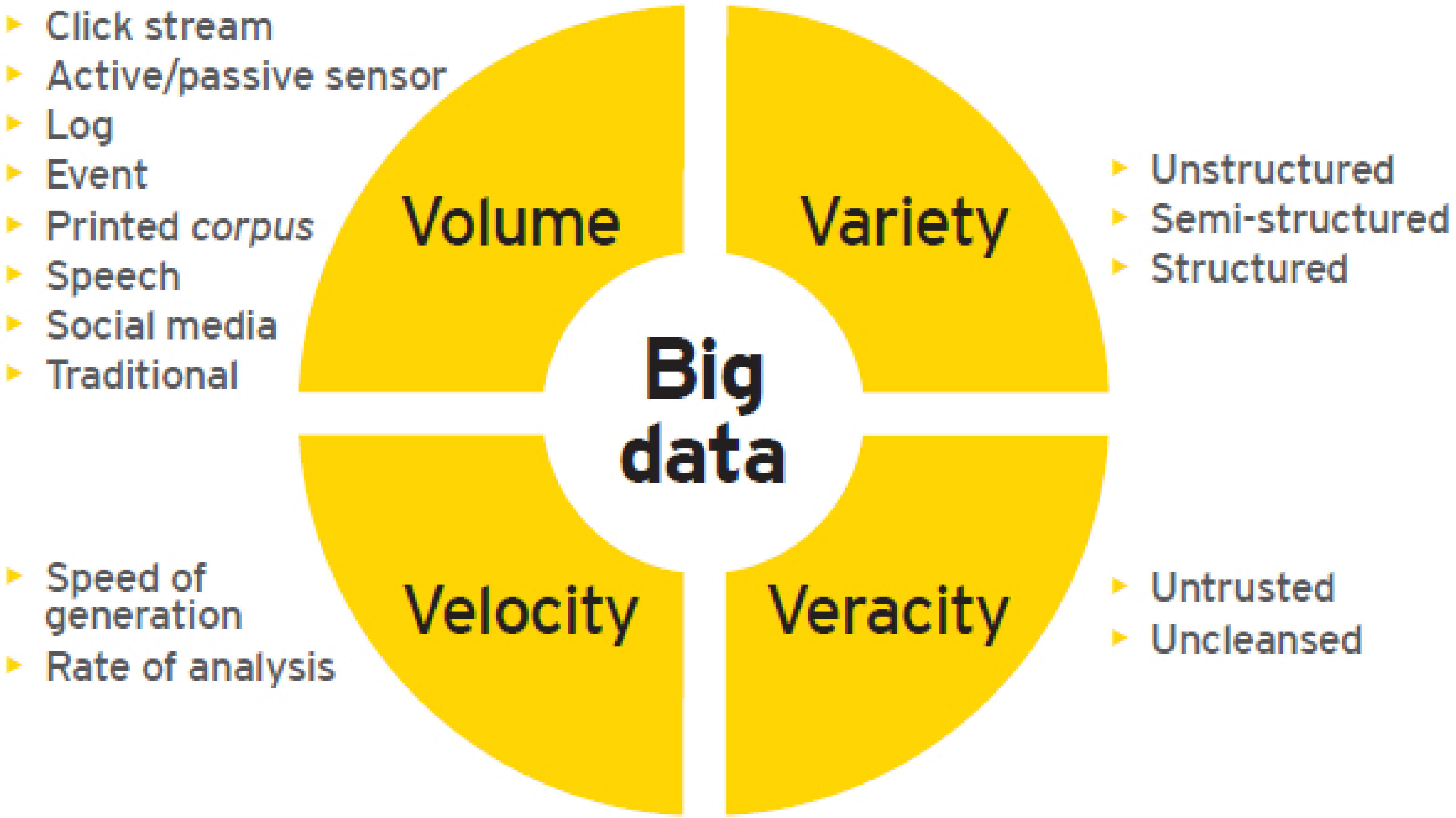
Velocity

- Data in Motion
- Analysis of streaming data

Veracity

- Data in doubt
- Uncertainity, inconsistency, incompleteness

4 V's of Big Data



Big data

Volume

- Click stream
- Active/passive sensor
- Log
- Event
- Printed *corpus*
- Speech
- Social media
- Traditional

Variety

- Unstructured
- Semi-structured
- Structured

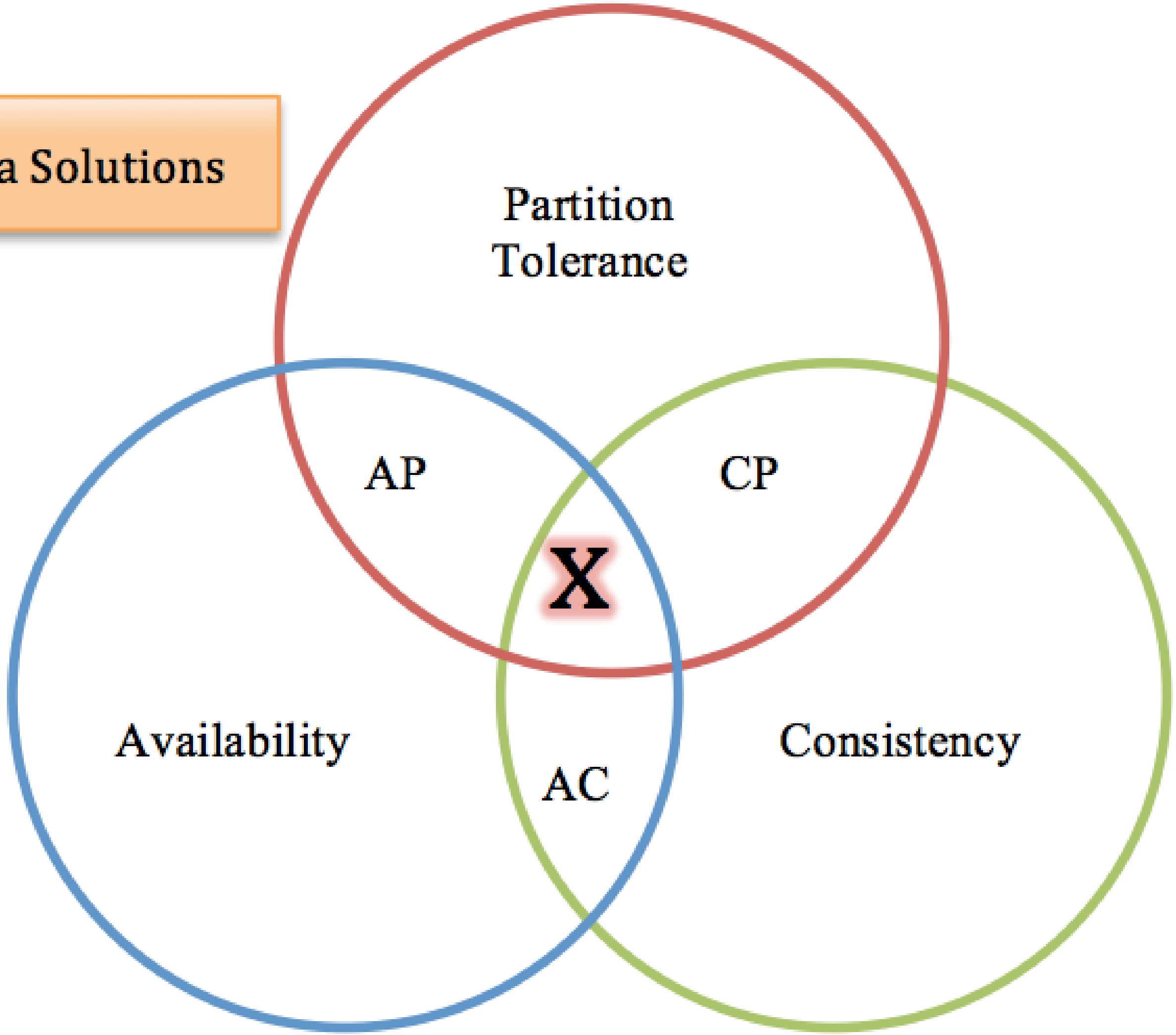
Velocity

- Speed of generation
- Rate of analysis

Veracity

- Untrusted
- Uncleansed

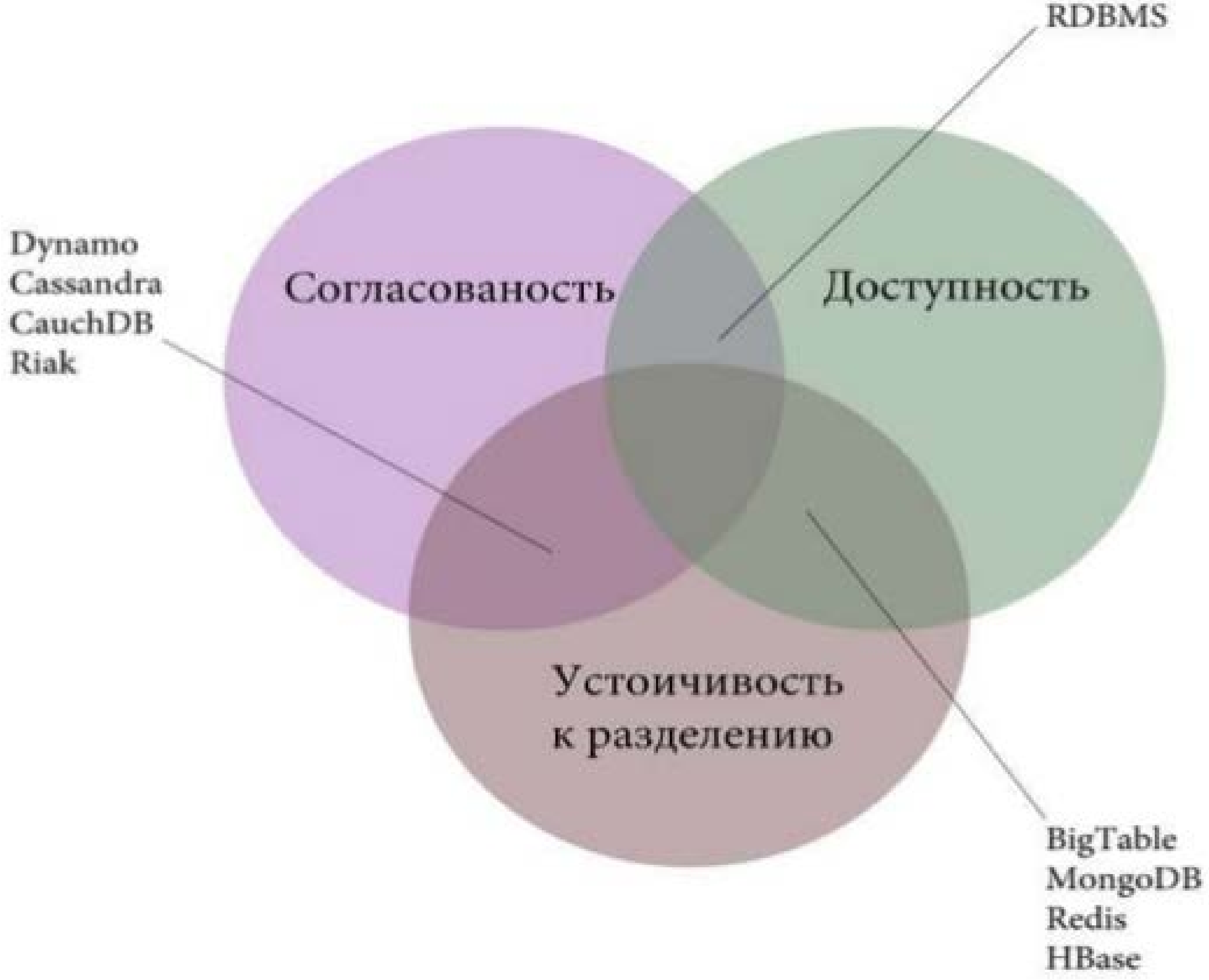
Big Data Solutions



CAP theorem as the way out

1. CAP as the description of the situation
2. CAP statement as the way to mark a line
3. New definition of Big DATA

CAP theorem data realizations



Conclusions

1. There are at least three different types of BIG DATA
2. We need an instrument to see where we are
3. DATA API is urgently needed
4. BIG DATA Ecosystem is a general solution
5. Software stack is a key question
6. The pack of tests is needed to get the answers

DATAAPIs

API is a business capability delivered over the Internet to internal or external consumers

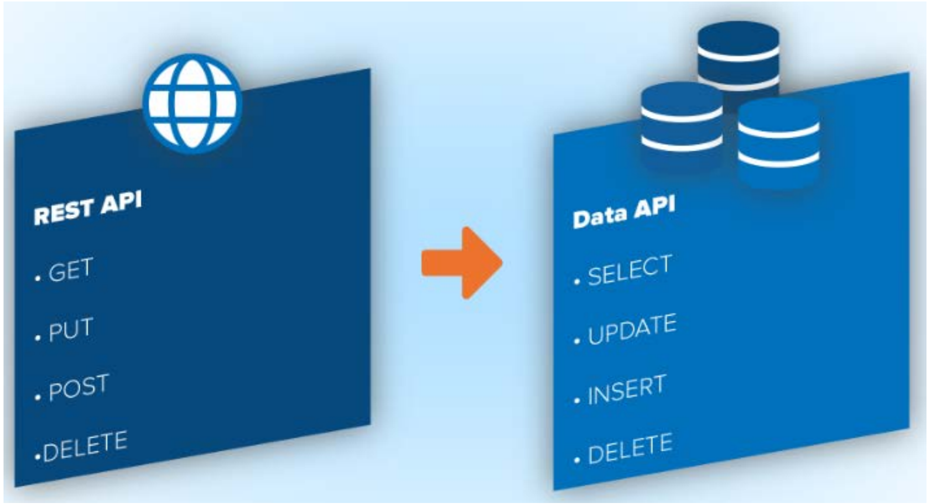
- Network accessible function
- Available using standard web protocols
- With well-defined interfaces
- Designed for access by third-parties

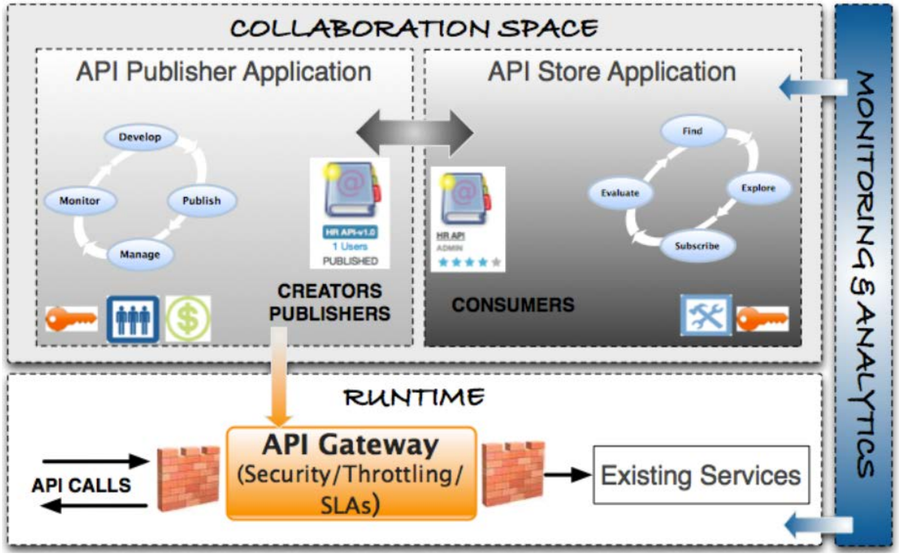
Managed API is:

- Actively advertised and subscribe-able
- Available with SLAs
- Secured, authenticated, authorized and protected
- Monitored and monetized with analytics

Data API: Unified approach to data integration

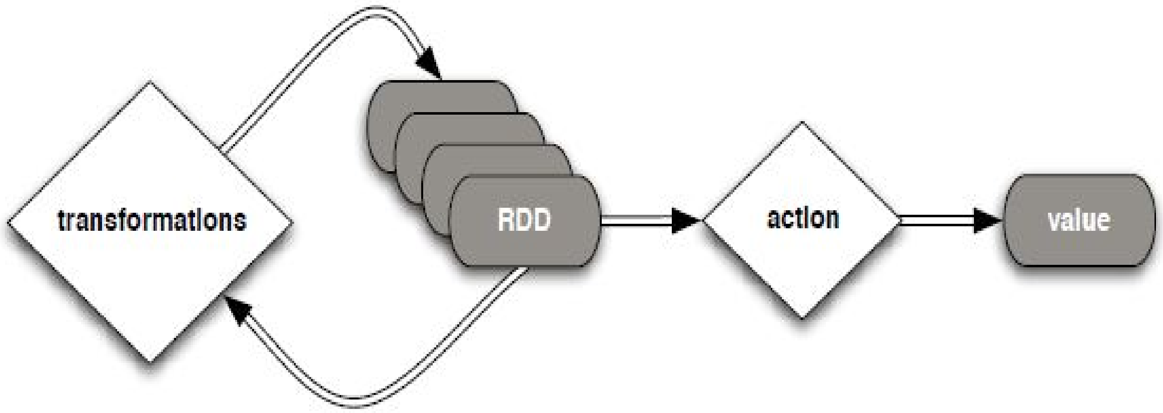
- Conventional APIs: Web, Web Services, REST API – not built for analytics
- Database paradigm: SQL, NoSQL, ODBS and JDBC connectors – familiar to analysts
- Database Metaphor + API = Data API
- Specific API for every type of big data (every “V” and their combinations) – under a generic paradigm





- Hadoop Distributed File System (HDFS)
- Hadoop YARN –including framework MapReduce.
- Hadoop common

Spark Lightning-fast cluster computing



Transformations vs operations

NoSQL

Документо-ориентированные БД	Граф-ориентированные СУБД
Couchbase MarkLogic mongoDB	Neo4j InfiniteGraph The Distributed Graph Database
Хранилища типа ключ-значение	Колоночные СУБД
redis amazon DynamoDB AEROSPIKE riak	accumulo HYPERTABLE INC Cassandra APACHE HBASE Amazon SimpleDB

Definition

The data is considered *big*, if its pre- and post-processing1 time is much larger than processing time.

Big data does not always mean big volume.

I Tightly-coupled data is big.

I High-volume data is big.

I Semi-structured data is big.

Edge cases.

I OpenFOAM: *tpre* + *tpost* _ *tproc* (not so big data)

1general I/O, decompressing, decoding, filtering etc.

Data metrics

Approach: Try to be as close as possible to the edge case (i.e. decrease pre/post time).

Tread -→ 0;

Twrite -→ 0:

No. of replicas/chunks:

I Capped by physical constraints (total no. of nodes, max. no. of nodes per job etc.)

The implementation

A lightweight Linux service.

I Portable C++ programme (8130 SLOC).

I Basically a scheduler that allows applications
to interact with a whole cluster via C++ API.

I An application for determining where file replicas are stored.

I An application for auto-discovery and building
virtual tree of healthy nodes

I An application for exposing basic web interface

.

Comparison to Hadoop

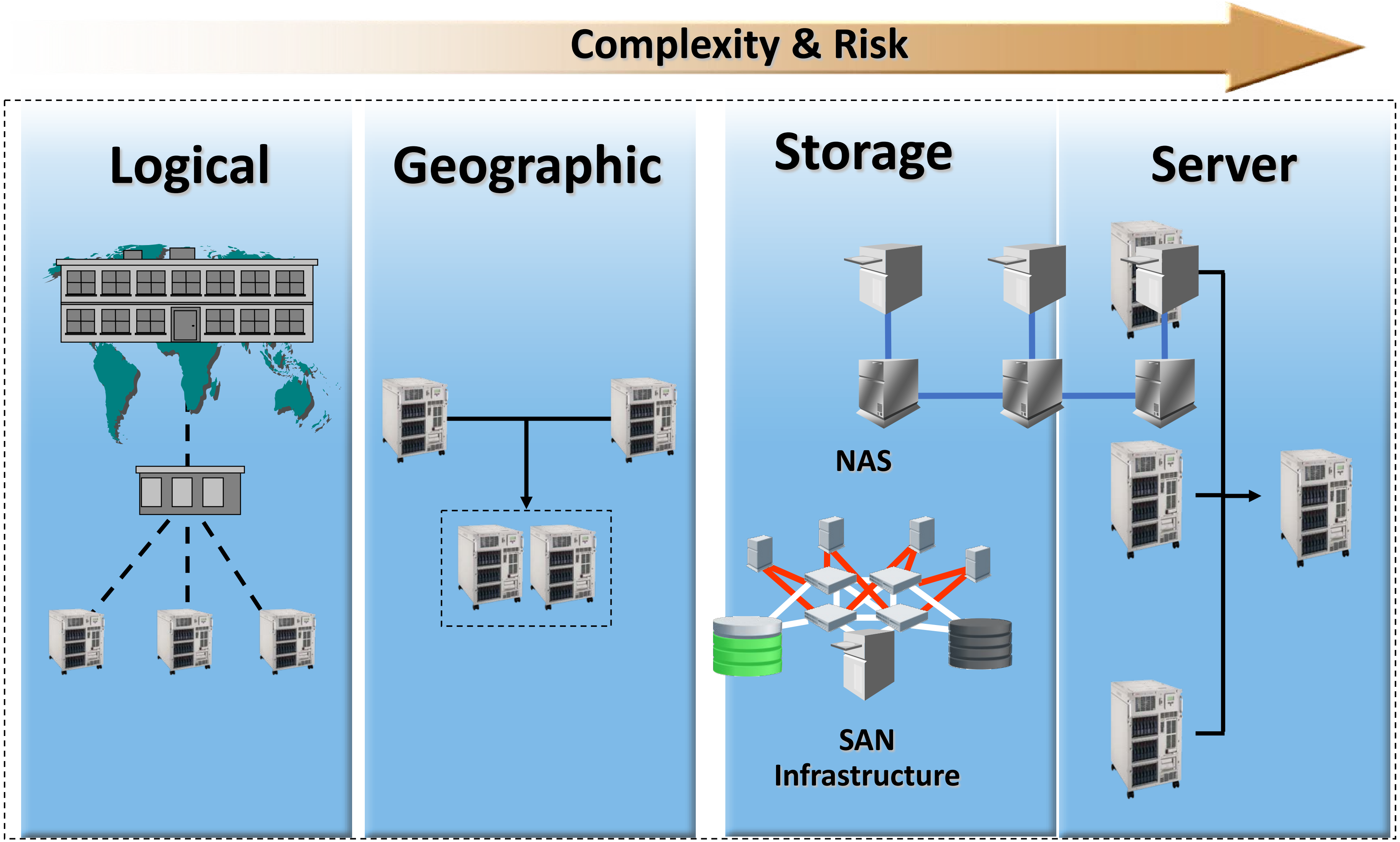
Setup

Hadoop version 2.3.0
Hadoop nodes 3
RAM (GB) 4
CPU Intel Q9650
No. of cores 4
Core freq. (GHz) 3.0
OS Debian 7.5

Performance

Hadoop 1000 spec./sec.
Factory 7000 spec./sec.

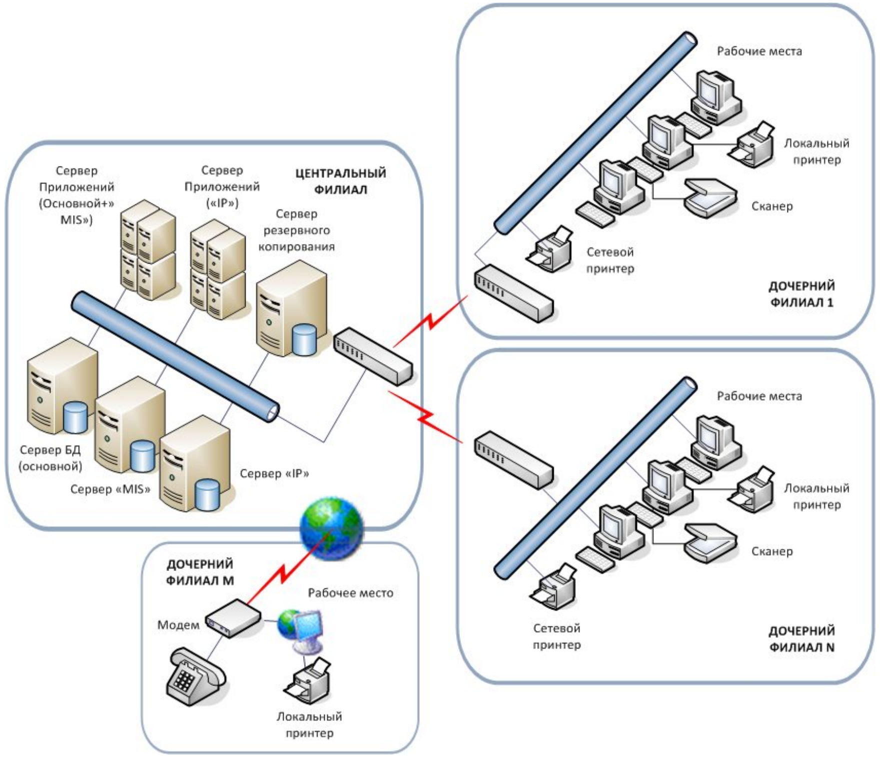
Consolidation Strategy

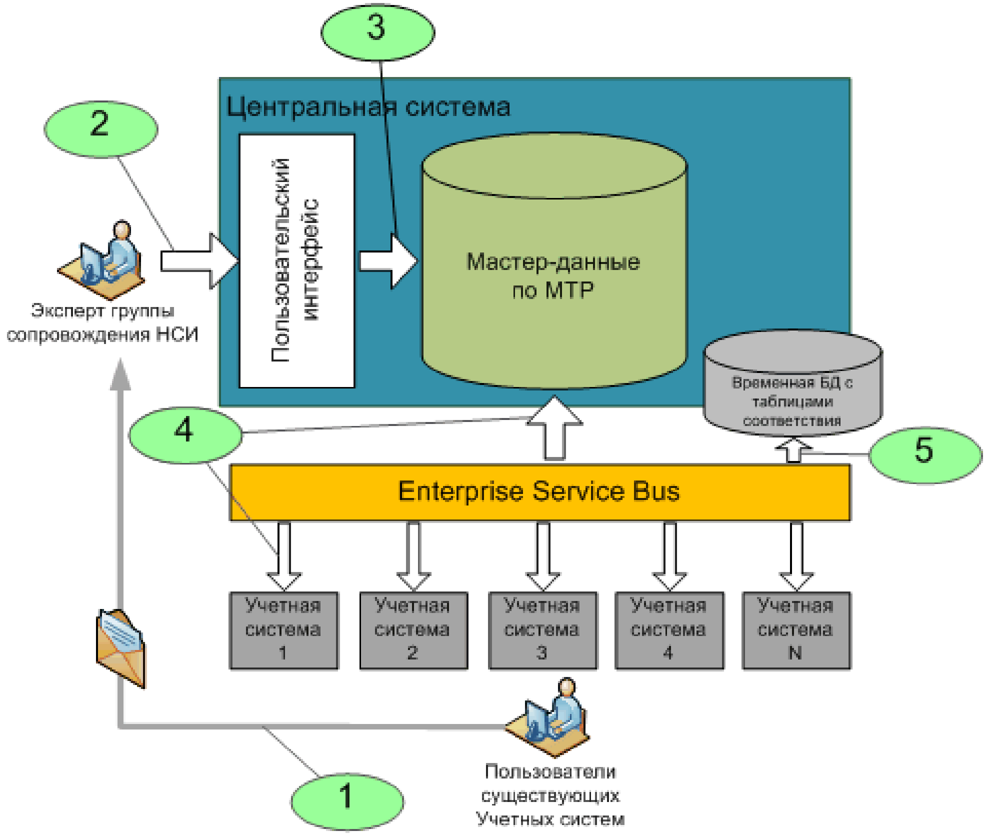


What Is Server Consolidation?

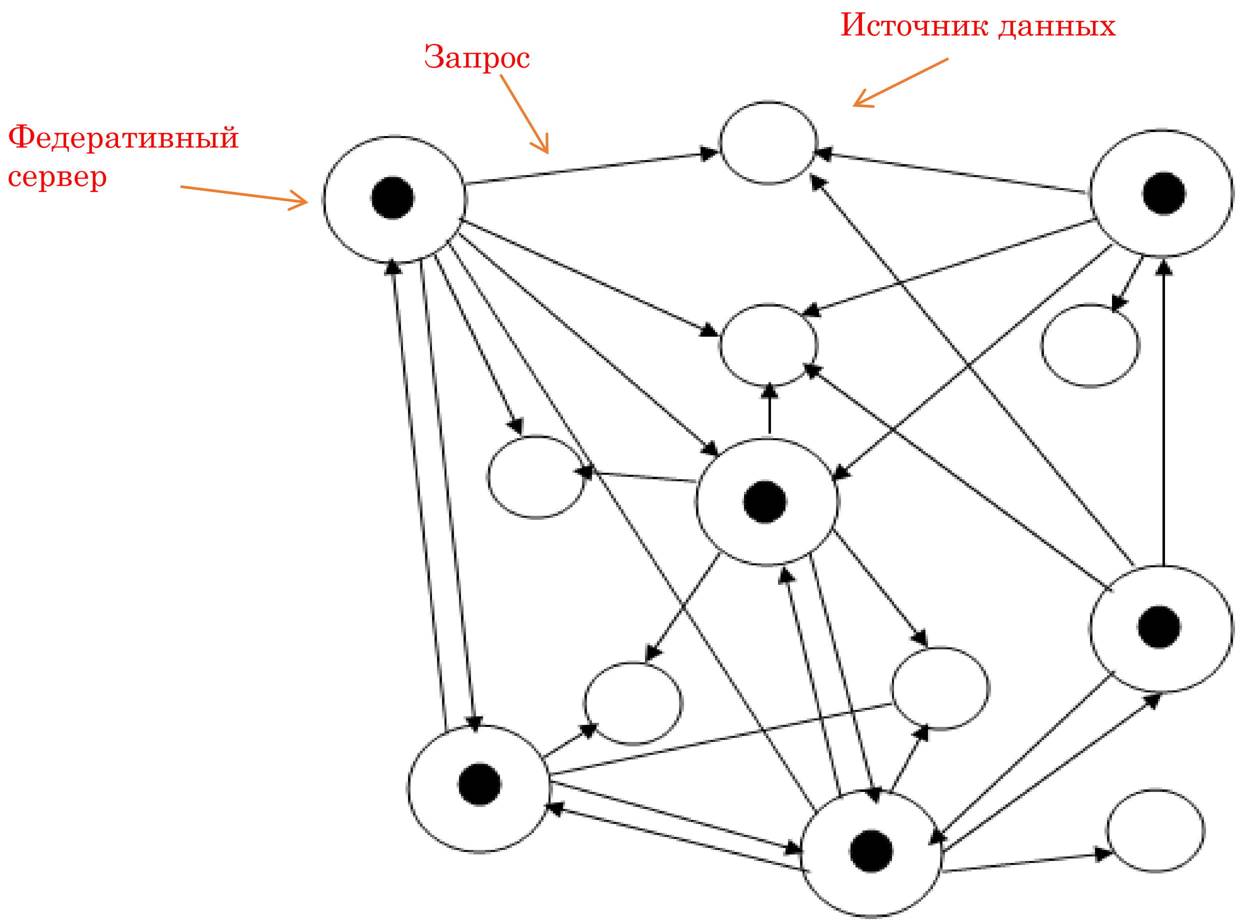
- Centralizing management of business-critical systems and applications
- Consolidation of applications onto fewer, more highly-available servers
- Standardization of platforms and processes
- Optimization of human and physical resources
- Consolidation of servers into one geographical location

Centralized DB architecture

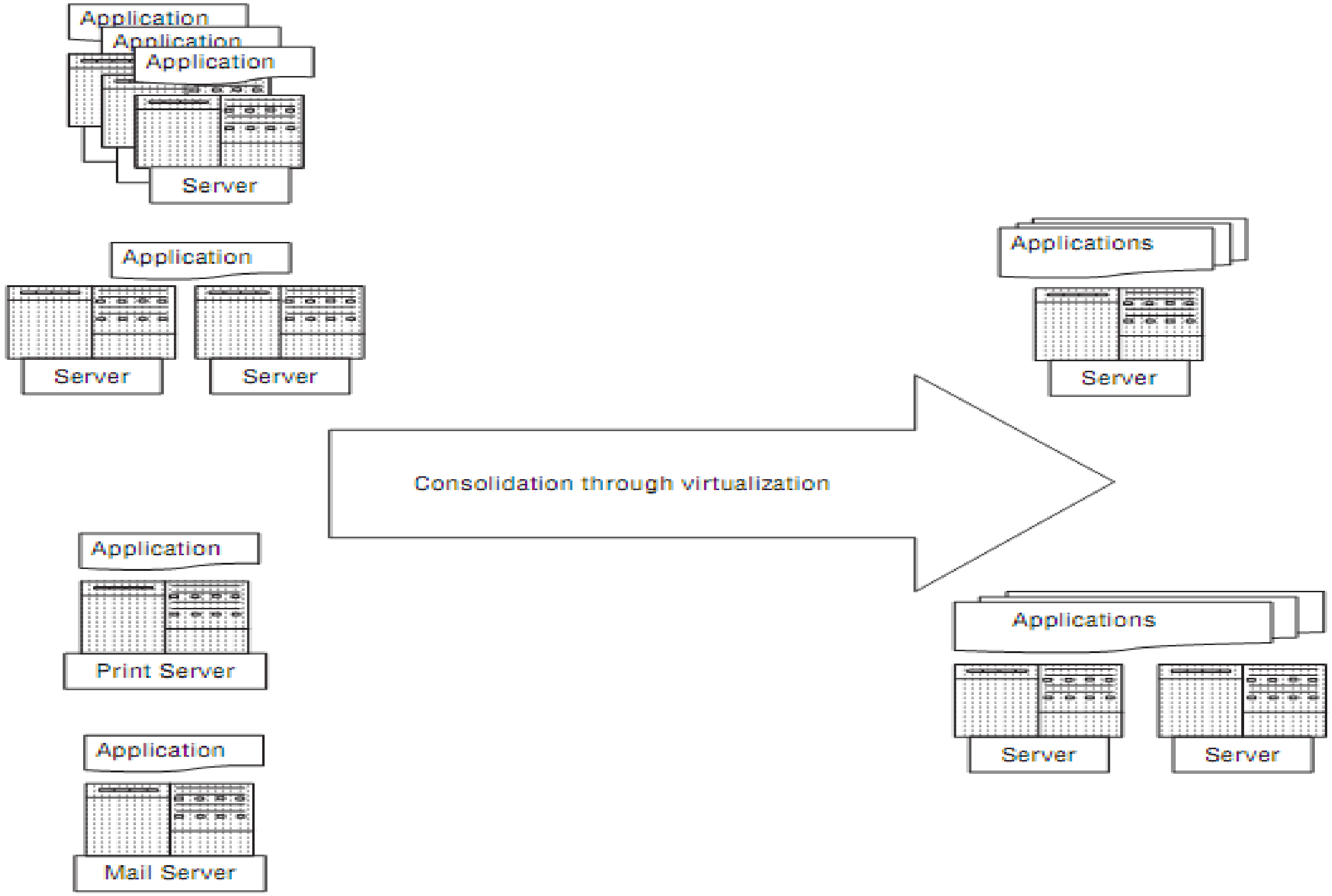




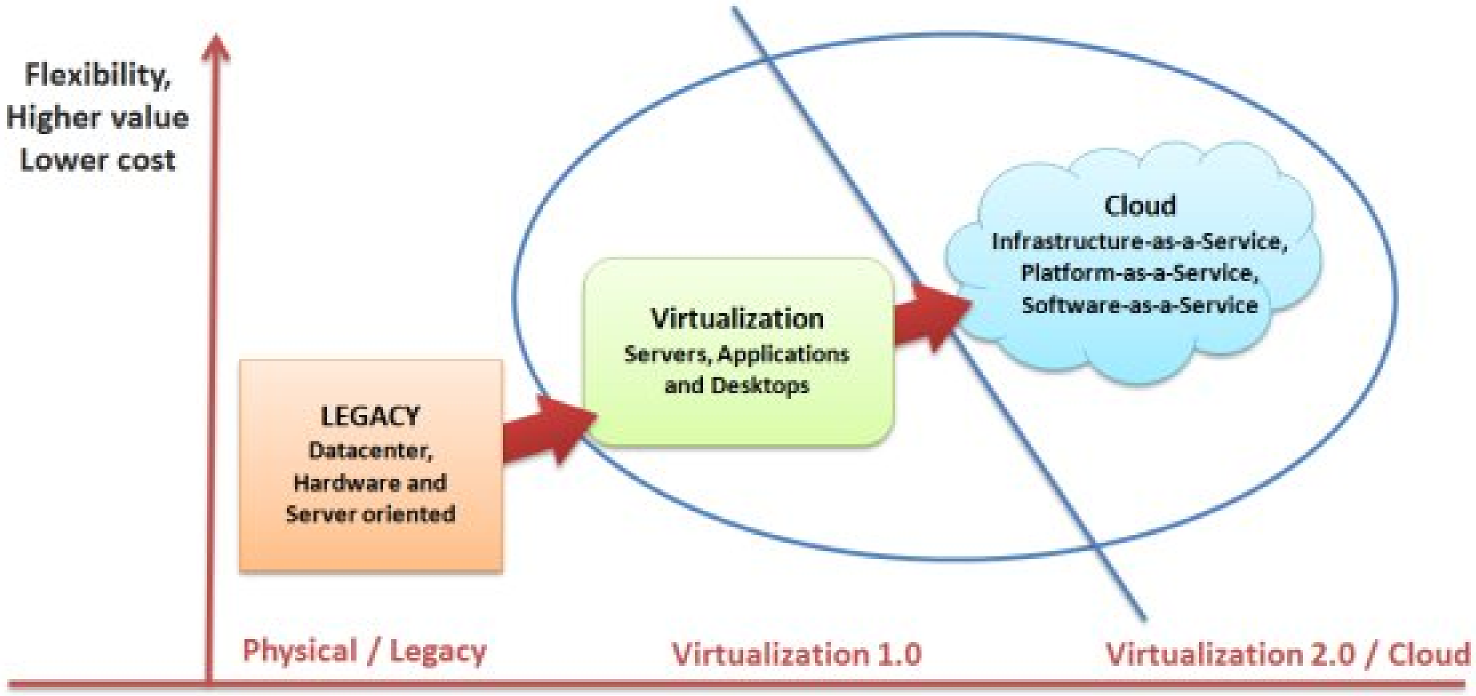
Federative DB architecture



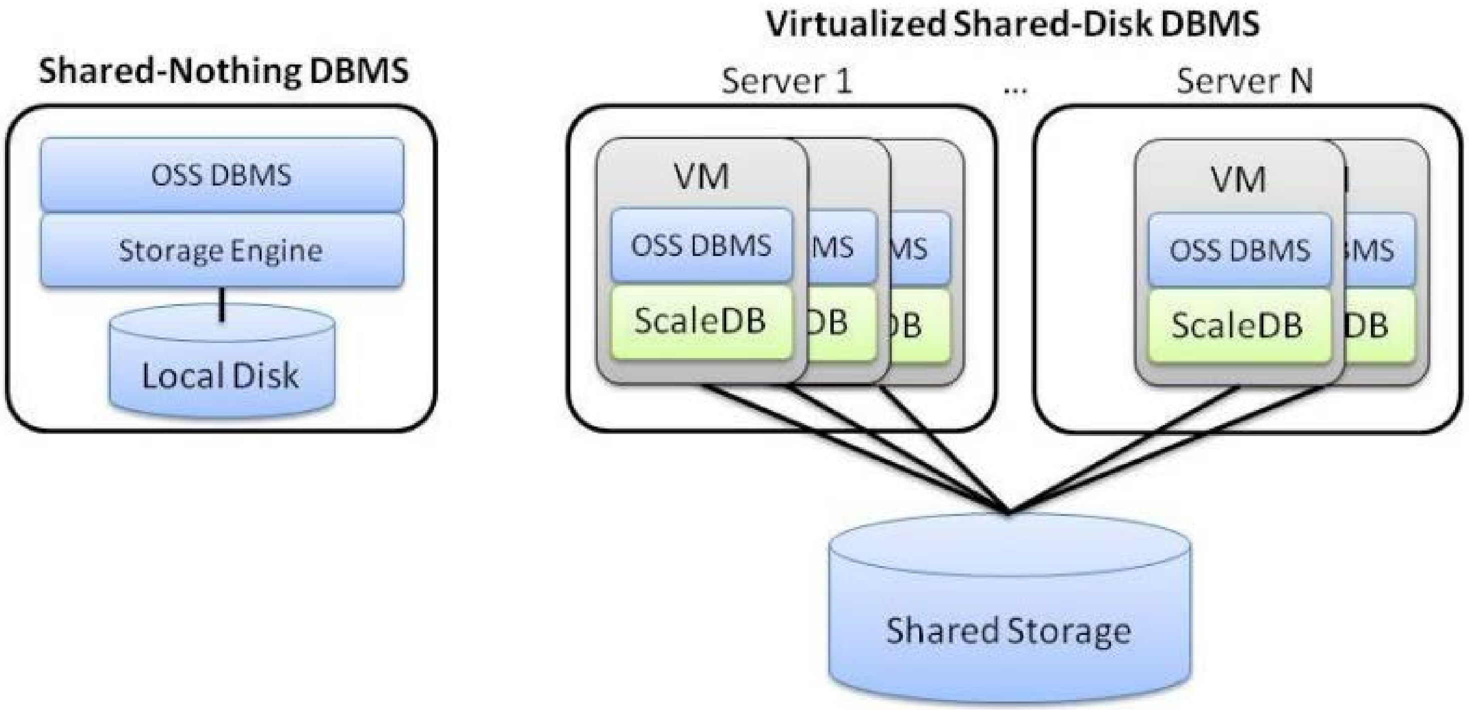
Server consolidation through virtualization



Cloud Computing & Virtualization



Database Virtualization and the Cloud



What is the difference of Cloud DBMS and DbaaS

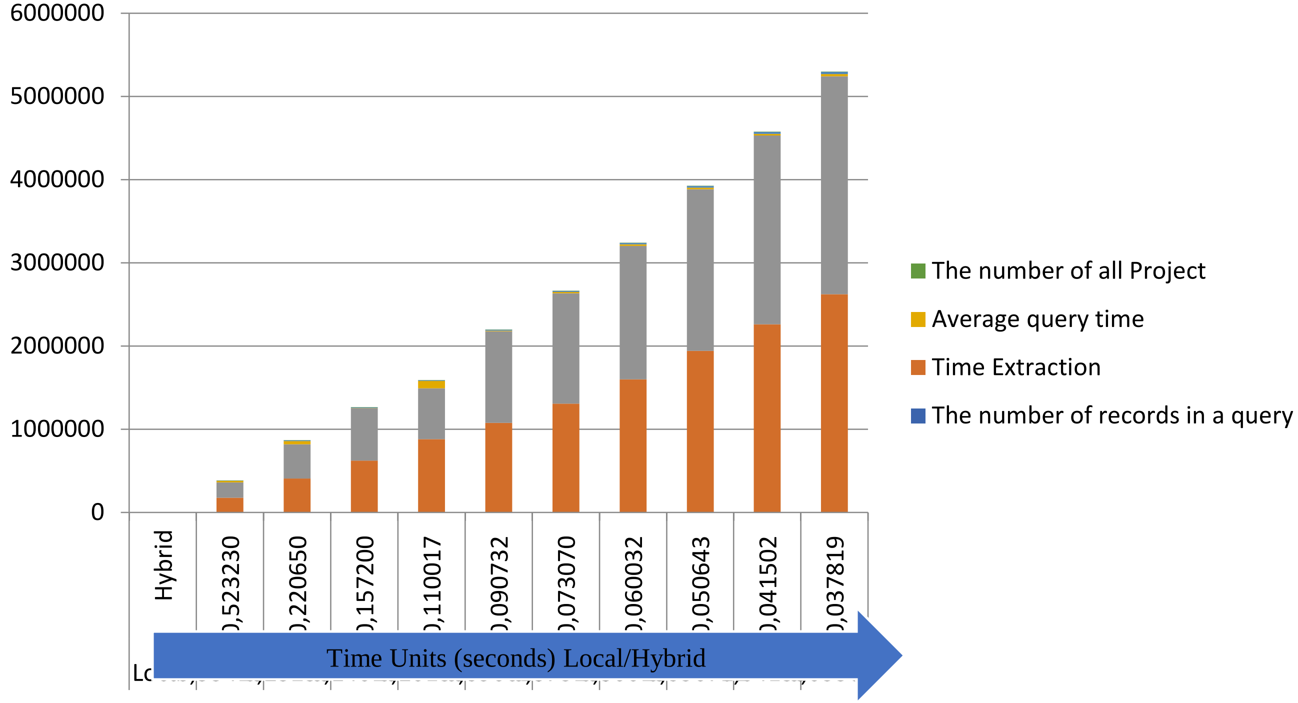
Cloud DBMS - is a fully automated multi-user and unlimited scalable service that provides database functionality, but operated and administered " unnoticed " by the service provider. It should not be confused cloud DBMS (this service) and database running on a virtual machine.

Database as a Service - providing a simple but functional profile of saturated solutions " database in the cloud " for the needs of medium and small businesses and IT departments of large corporations . It usually does not occur directly in the provider 's own data center, and functions as an add-on classical cloud services. Almost always specific DbaaS - this one particular database provided in the cloud directly to the developer.

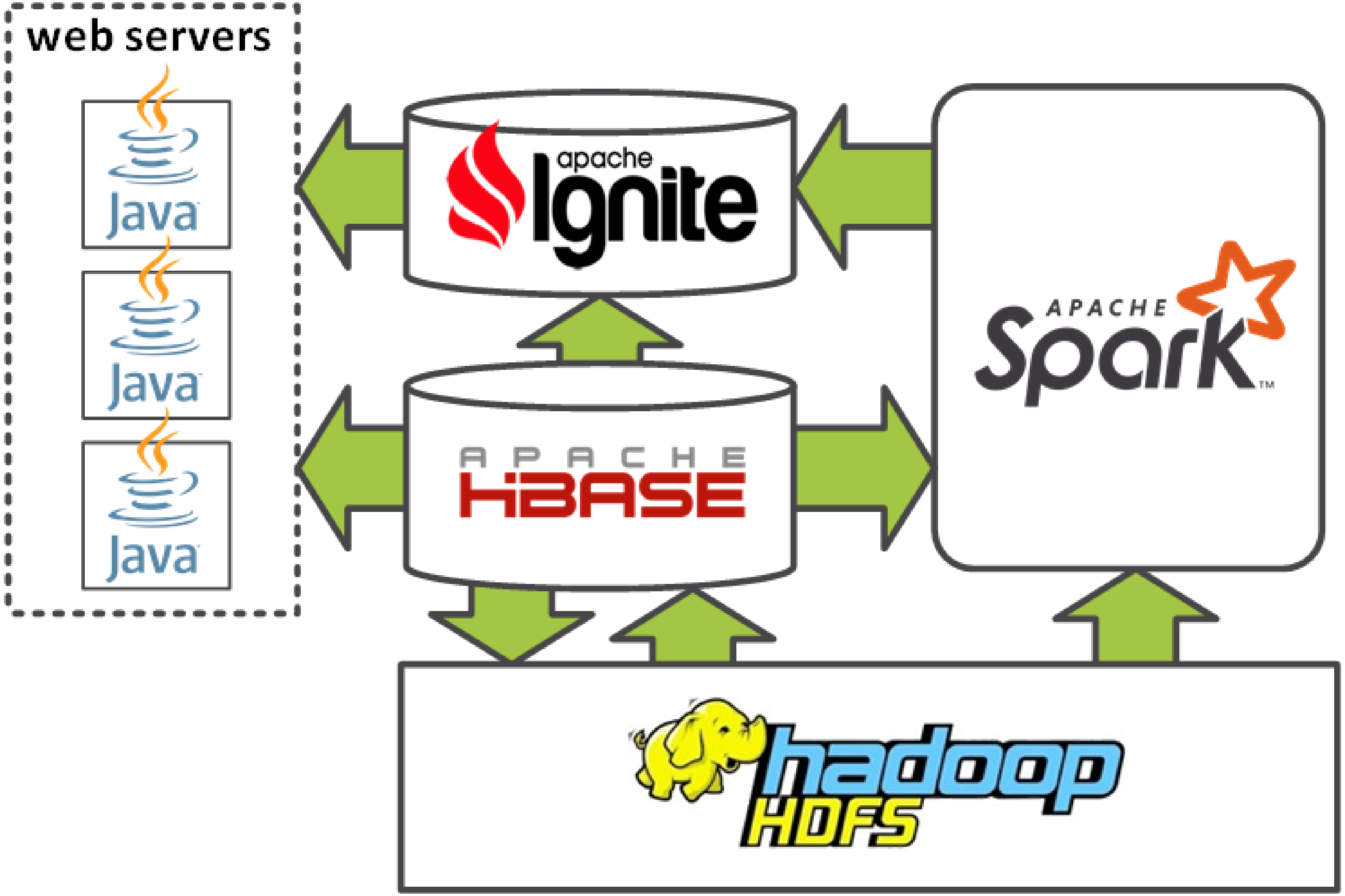
Analysis of the Database and Hybrid Local Time Condition

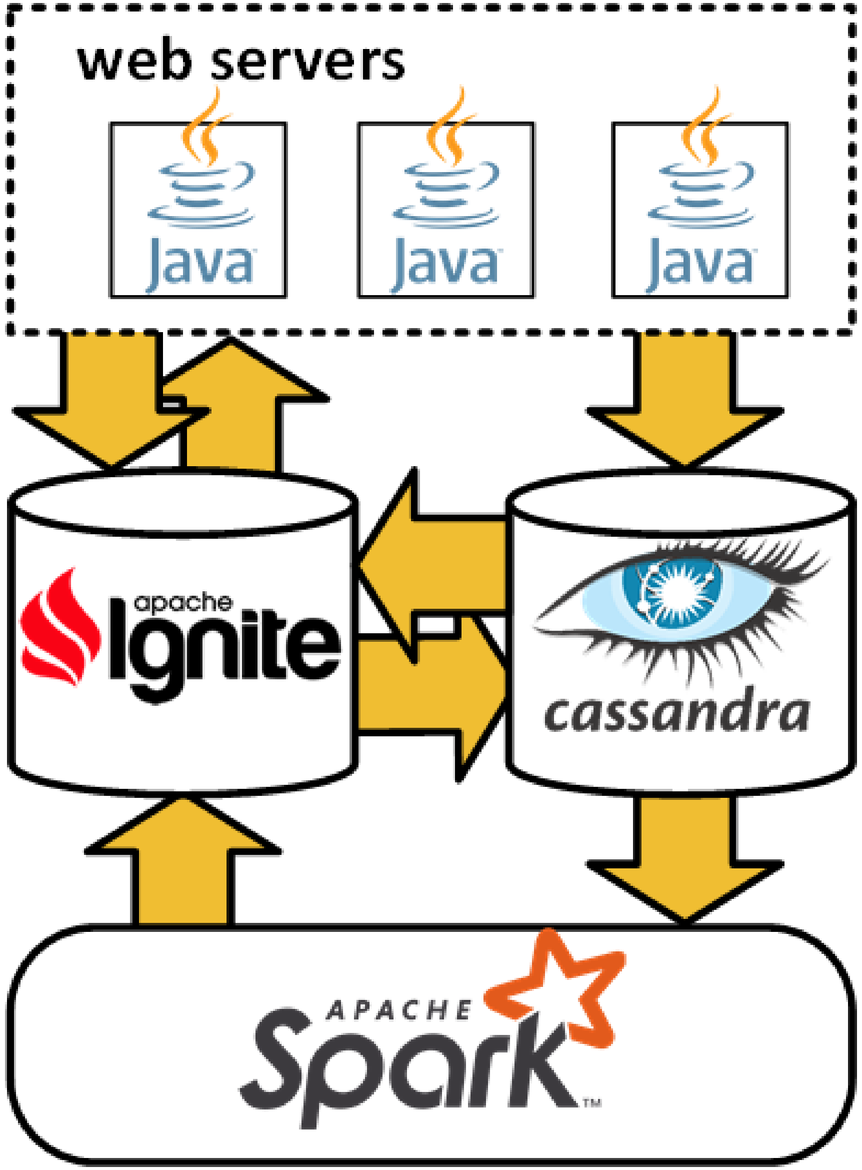
Record of Seconds		The number of records in a query	Time Extraction		Average query time		The number of all Project
Local	Hybrid		Local	Hybrid	Local	Hybrid	
0,534101	**0,523230**	**1**	**177,533**	**183,334**	**1,7975**	**1,833**	**100**
0,232501	**0,220650**	**2**	**406,403**	**411,739**	**4,0862**	**4,117**	**200**
0,140102	**0,157200**	**3**	**621,604**	**627,270**	**6,238**	**6,272**	**300**
0,101307	**0,110017**	**4**	**880,467**	**883,8712**	**8,8168**	**8,838**	**400**
0,090013	**0,090732**	**5**	**1075,044**	**1099,451**	**10,960**	**10,994**	**500**
0,073105	**0,073070**	**6**	**1305,506**	**1329,938**	**13,055**	**13,077**	**600**
0,060137	**0,060032**	**7**	**1600,401**	**1604,110**	**16,014**	**16,041**	**700**
0,050722	**0,050643**	**8**	**1940,710**	**1943,777**	**19,407**	**19,437**	**800**
0,04155	**0,041502**	**9**	**2262,453**	**2265,402**	**22,624**	**22,653**	**900**
0,03565	**0,037819**	**10**	**2620,356**	**2622,060**	**26,203**	**26,220**	**1000**

Experiment results Graph



Time Units (seconds) Local/Hybrid


web servers
Java
Java
Java
apache Ignite
APACHE HBASE
APACHE Spark
hadoop HDFS


web servers
Java
Java
Java
apache Ignite
cassandra
APACHE Spark

What are the best methods for testing big data applications?

Step 1: Data Staging Validation

Data from various source like RDBMS, weblogs etc. should be validated to make sure that correct data is pulled into system.

Comparing source data with the data pushed into the Hadoop system to make sure they match.

Verify the right data is extracted and loaded into the correct HDFS location

Step 2: "Map Reduce"Validation

Map Reduce process works correctly

Data aggregation or segregation rules are implemented on the data

Key value pairs are generated

Validating the data after Map Reduce process

Step 3: Output Validation Phase

To check the transformation rules are correctly applied

To check the data integrity and successful data load into the target system

To check that there is no data corruption by comparing the target data with the HDFS file system data

Architecture Testing

Performance Testing

1. Data ingestion and Throughout
2. Data Processing Sub-Component Performance

What a user should do?

1. Estimate the total system parameters (maximum number of users for simultaneous operation , the ability to scale services, the availability of personalized access).

2 . Evaluate the the project (having our own server capacity, cost comparison with the cost of building rental services) .

3 . Evaluate time data access, query performance evaluation for cloud infrastructures.

4 . Construct the automatic allocation system and send requests in a distributed database.

Conclusions

- 1. New definition works
- 2. DATAAPI is “must have”
- 3. Future is DATA Ecosystem
- 4. Large amount of tests is still needed
- 5. BD measure should be more detailed

Thank you for attention!

Ready for questions