Úvod do Big Data

Martin Macák

Fakulta informatiky, Masarykova univerzita, Brno

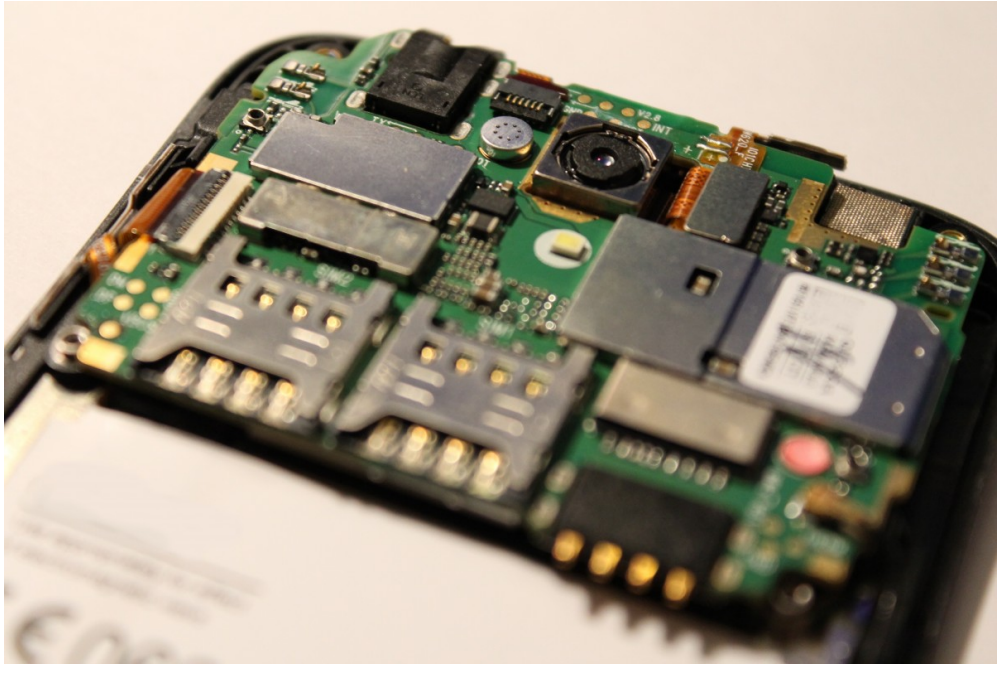
21. 2. 2019

Osnova prednášky

1. Čo sú to Big Data?
2. Ako pracovať s Big Data?
3. Aké nástroje existujú?

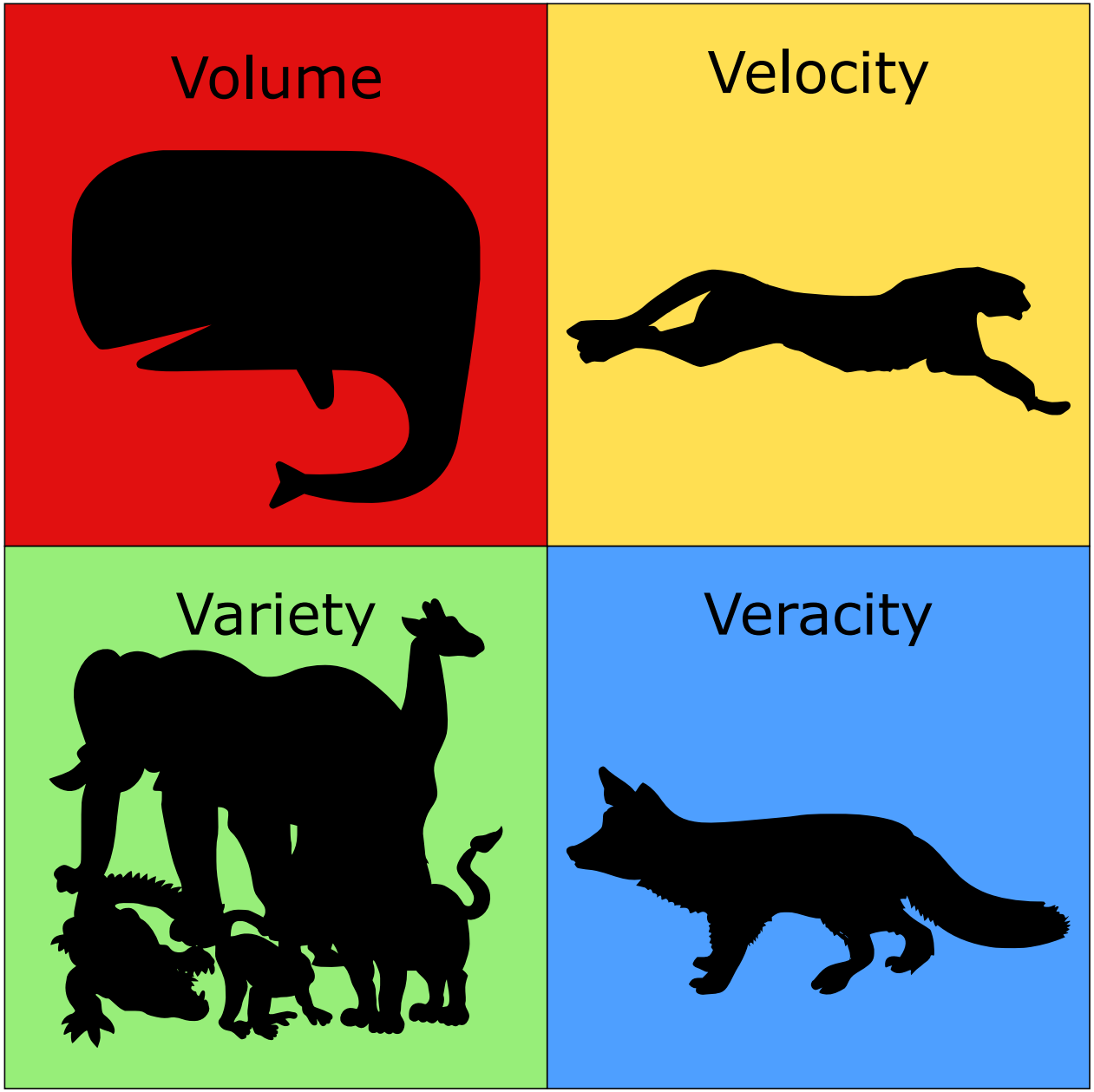
Big Data

- *data is the new oil*
- všetci generujeme dáta
- **koľko senzorov má napr. smartphone?**



- problem nie je generovať dáta, ale ich spracovať!

Big Data – definícia



Typické základné požiadavky na Big Data systémy

- ukladanie veľkého množstva dát
- spracovanie dát v rozumnom čase
- škálovateľnosť

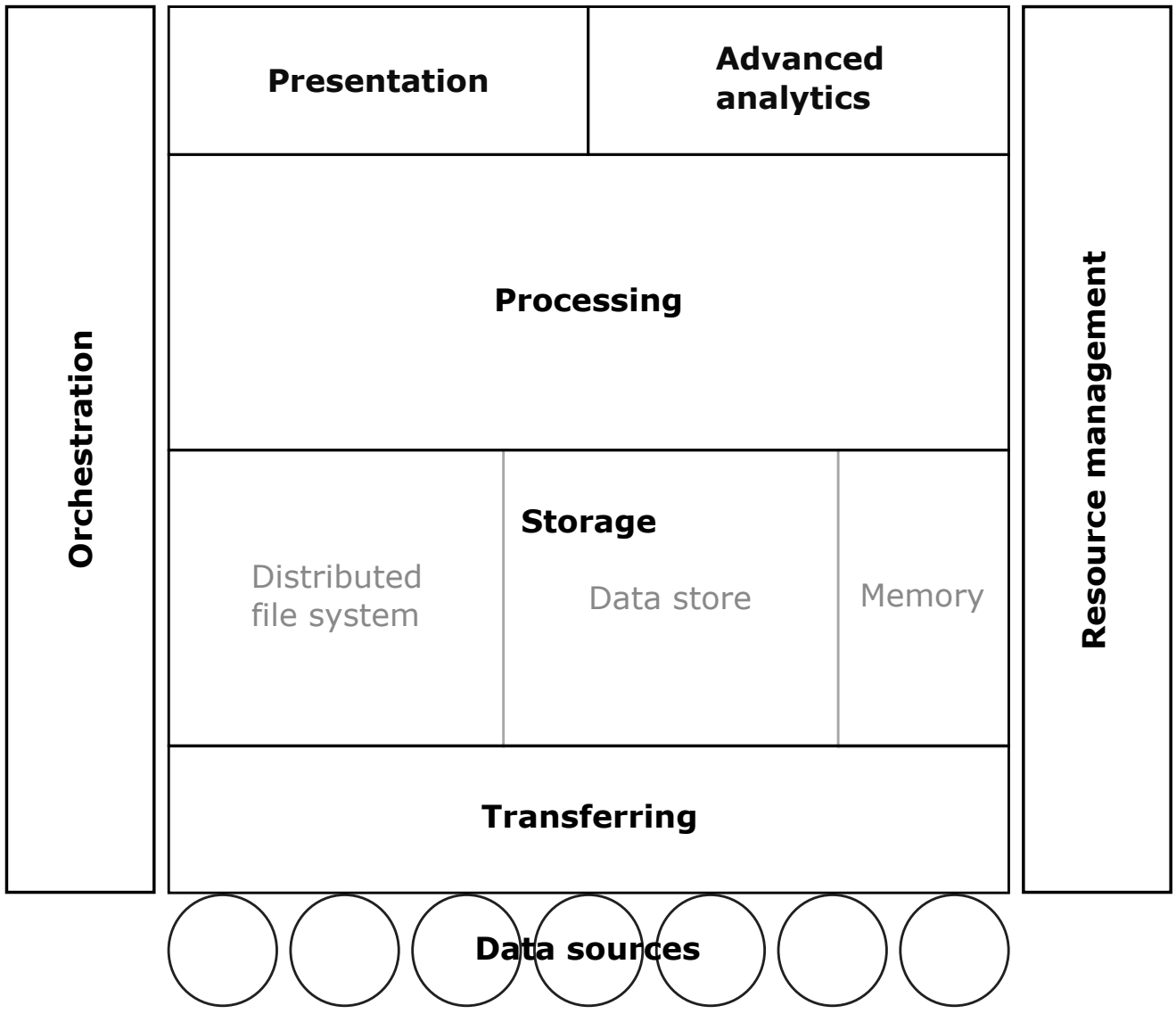
Big Data systémy – superpočítač

- vertikálne škálovanie

Big Data systémy – cluster bežných počítačov

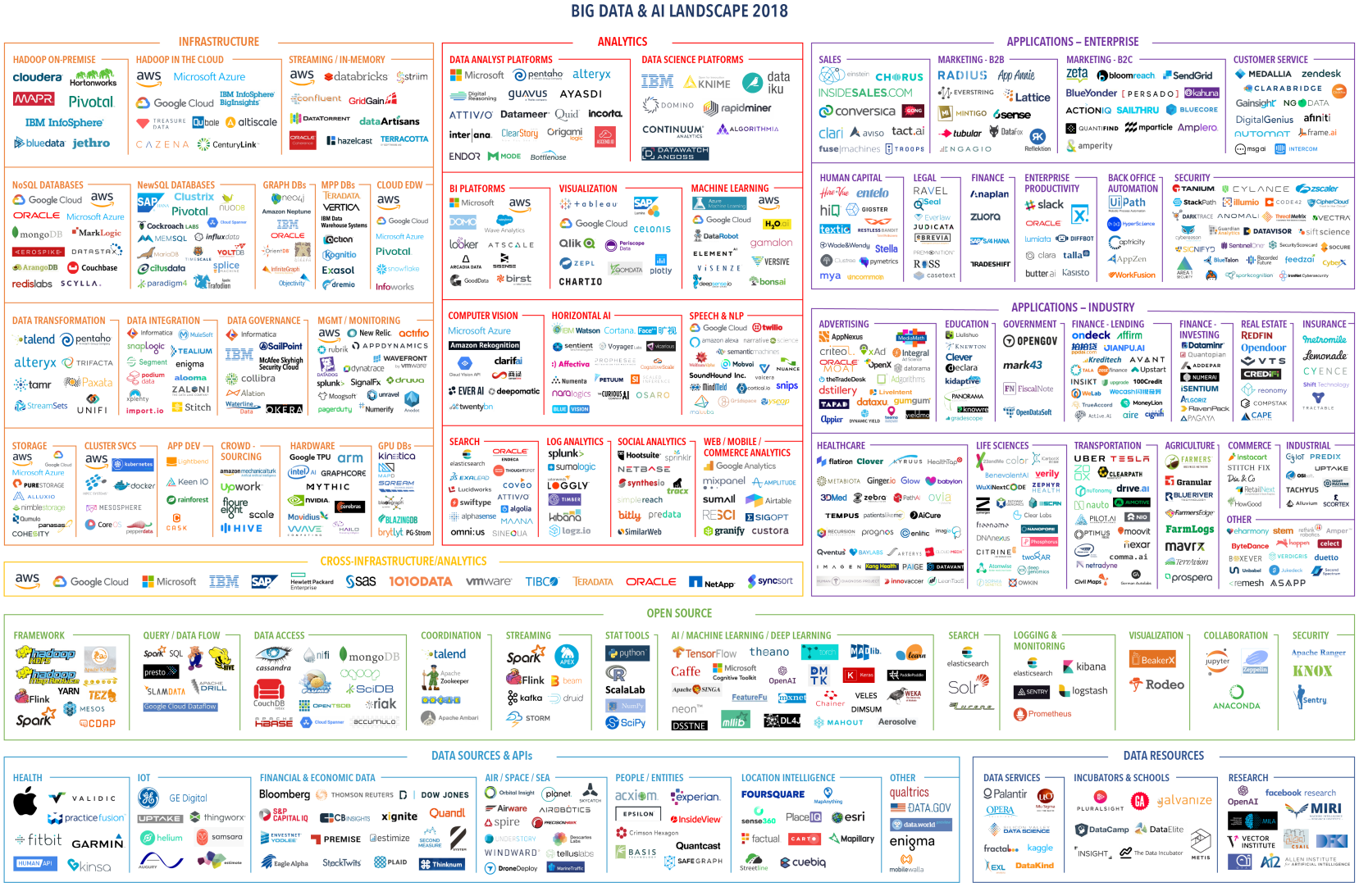
- horizontálne škálovanie

Typy práce


Orchestration
Presentation
Advanced analytics
Processing
Storage
Distributed file system
Data store
Memory
Transferring
Resource management
Data sources

Nástroje

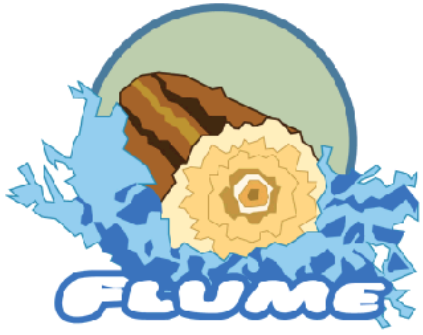
- pre každý typ práce existuje mnoho nástrojov



FIRSTMARK
EARLY STAGE VENTURE CAPITAL

Transferring

- presun dát zo zdrojov (zariadenia, databázy, súbory, webové stránky, . . .)
- aktívne alebo pasívne presúvanie
- väčšinou máme možnosť predspracovania dát (filtrovanie, transformácia)



Storage

- CAP Theorem
  - consistency – každý uzol má rovnaký pohľad na dáta, vždy vráti najnovší úspešný zápis
  - availability – každý uzol vráti odpoveď v rozumnom čase
  - partition tolerance – systém dokáže pracovať aj pri výpadkoch
- v distribuovanom prostredí proti sebe "bojujú" *consistency* a *availability*
- nemôžeme zároveň garantovať, že vždy vrátime odpoveď, a že vrátená informácia je správna

Storage

- Distributed File Systems
  - súbory

- Relational Database Management Systems
  - štruktúrované dáta
  - SQL

Storage – NoSQL Database Management Systems

Nepotrebujú preddefinovanú schému

- Key-Value stores
  - typicky operácie put, get, delete
  - rýchle

- Document stores
  - Key-Value, kde Value má štruktúru (JSON, XML, . . .)
  - možnosť komplexnejších dotazov

```
db.users.find( { name: "Martin" } )
```

Storage – NoSQL Database Management Systems

- Column-family stores
  - column families s riadkami
  - každý riadok je Key-Value, kde Value je množina stĺpcov (meno-hodnota)
  - dotazovanie podobné SQL

- Graph databases
  - ukladajú vrcholy a hrany medzi nimi (môžu mať atribúty)
  - dotazovanie typicky cez jazyk Cypher alebo Gremlin

```
MATCH (martin:Person {name:"Martin"})
  -[:FRIEND]-(mutualFriend:Person)
  -[:FRIEND]-(tomas:Person {name:"Tomas"})
RETURN mutualFriend
```

Storage

- Multi-model databases
  - uľahčujú tzv. *polyglot persistence*
  - viacero typov databáz v jednej
  - dotazovanie podobné SQL

- špecializované databázy
  - Time-series, Spatial, ...
  - táto funkcionalita môže byť už zahrnutá v iných databázach

Processing

Výpočty musia byť paralelizovateľné!

- dávkové spracovanie

- prúdové spracovanie

- grafové spracovanie

- vysokoúrovňovejšie spracovanie

- všeobecne účelové spracovanie

Apache Flink

Processing - Hadoop MapReduce

```
class WordCountMapper extends Mapper[Object,Text,Text,IntWritable]
{
  override
  def map(key:Object, value:Text,
    context:Mapper[Object,Text,Text,IntWritable]#Context) =
  {
    value.toString().split("\\W+")
      .map(word => context.write(new Text(word), new IntWritable(1)))
  }
}

class WordCountReducer extends Reducer[Text,IntWritable,Text,IntWritable]
{
  override
  def reduce(key:Text, values:java.lang.Iterable[IntWritable],
    context:Reducer[Text,IntWritable,Text,IntWritable]#Context) =
  {
    val sum = values.foldLeft(0) { (t,i) => t + i.get }
    context.write(key, new IntWritable(sum))
  }
}
```

Processing - Giraph

```
public void compute(Iterator<DoubleWritable> msgIterator)
{
  if (getSuperstep() == 0)
  {
    setVertexValue(new DoubleWritable(Double.MAX_VALUE));
  }
  // if you are a source vertex, set min to 0, else infinity
  double min = (getContext().getConfiguration()
      .getLong(SOURCEID, SOURCEIDDEFAULT)
        ==  getVertexId().get()) ? 0d : Double.MAX_VALUE;
  while (msgIterator.hasNext())  // read all received messages
  {
    min = Math.min(min, msgIterator.next().get());
  }
  if (min < getVertexValue().get())
  {
    setVertexValue(new DoubleWritable(min));
    // send a message to all neighbors
    for(Edge<LongWritable, FloatWritable> edge
        : getOutEdgeMap().values())
    {
      sendMsg(edge.getDestVertexId(),
        new DoubleWritable(min + edge.getEdgeValue().get()));
    }
  }
  voteToHalt();  // set to inactive
}
```

Processing - Hive

```
SELECT word, count(1) AS count FROM
 (SELECT explode(split(cities, ' '))
   AS word FROM users)tempUsers
GROUP BY word
```

Processing - Spark

```
val counts = textFile
  .flatMap(_.split("\\W+"))
  .map(_, 1)
  .reduceByKey(_ + _)
```

Rekapitulácia

1. Čo sú to Big Data?
   - new oil
   - 4V
   - veľa aplikačných domén
2. Ako s nimi pracovať?
   - superpočítač vs cluster
   - veeeeľa nástrojov
   - mnoho účelov
     - Transferring
     - Storage
     - Processing
     - ...

macak@mail.muni.cz