This seminar will use the EasyChair system to manage the reports and reviews.

The submission website is https://easychair.org/conferences/?conf=sbdm 2016

The deadline of the first version of the report is 7 Mar, 2016.
Log in to EasyChair for SBDM2016

EasyChair uses cookies for user authentication. To use EasyChair, you should allow your browser to save cookies from easychair.org.

User name: 
Password: 

Log in

If you have no EasyChair account, create an account
Forgot your password? click here
Problems to log in? click here
Information when you prepare your presentation

Seminar on big data management

Lecturer: Jiaheng Lu

Spring 2016
Your attitudes are important

- Are you INTERESTED in your topic?
  - If no, get a different one!
  - If yes, ACT LIKE IT
- If YOU are not excited …
  - Can not expect OTHER people to be!
Avoid dead man talking

- Are you hiding behind the podium?
- Are your hands/face motionless?
- Are you staring …
  - at your laptop
  - At the screen
  - At the ceiling
- IF SO … you are probably BORING
Is your back to the audience?
9. Do not read directly from your notes. Keep eye contact with your audience.
Right here. See?

- Do not point at your laptop screen
- They cannot see it
Practice is important

- Practice makes perfect (at least three times before your talk)
- Do not read your slides like a script
- Most people lose 20 IQ points in front of an audience
In general, do not have only text on most of your slides

Try to draw diagrams wherever applicable

(Well-drawn) pictures easier to understand
Anticipatory lecturing

- Do not be a tease

- Let the audience think at their own pace

- It only provides benefit if there is a ”surprise” result.
Results explanation

- You have lots of cool results
- No one can read this
- No one can understand this
- Graphs are your friend
Keep it simple

- Do you really need all those equations?
  - This is very instance-dependent
  - Depends on what you are discussing
  - Depends on your audience
- Sometimes you may need them
  - Explain the variables
    and what they mean
  - Given a "plain-text" description of it

\[ \nabla \cdot \mathbf{A} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z} \]

= (rate of change of \( A \) in x-direction) +
  (rate of change of \( A \) in y-direction) +
  (rate of change of \( A \) in z-direction)
Summay and conclusion

- Remember to summarize work and results

- Giving “selling” points here
  - 30X performance increase with only 10% area penalty
Before your presentation

- Please discuss with me about your presentation paper at least one week in advance of your presentation

- Please send the slides to the opponent and me before your presentation
Bad presentations
Wish your presentation to be a good presentation
New trends of big data management in 2016

Seminar on big data management

Lecturer: Jiaheng Lu

Spring 2016
Trend 1: Spark grows fast
Spark “on the radar”

- 2008 - Yahoo! Hadoop team collaboration with Berkeley Amp/Rad lab begins
- 2009 - Spark example built
- 2011 - “Spark is 2 years ahead of anything at Google”
  - Conviva seeing good results with Spark
- 2012 - Yahoo! working with Spark / Shark
- Today - Many success stories
  - Early commercial support
Spark updates Hadoop

- Hardware had advanced since Hadoop started:
  - Very large RAMs, Faster networks (10Gb+)
  - Bandwidth to disk not keeping up
- MapReduce awkward for key big data workloads:
  - Low latency dispatch (E.G. quick queries)
  - Iterative algorithms (E.G. ML, Graph…)
- Streaming data ingest
Spark, “lingua franca?”

- Support for many development techniques
  - SQL, Streaming, Graph & in memory, MapReduce
  - Write “UDFs” once and use in all contexts
- Small, simple & elegant API
  - Easy to learn and use; expressive & extensible
  - Retains advantages of MapReduce (fault tolerance…)
Spark often better

- Today you will hear many success stories from teams who have converted Hadoop based workloads to Spark and seen:
  - Huge speedups and Big cost savings
  - But there do exist cases where Hadoop is superior…
    - Proven to work at the largest scales
    - Mature & widely commercially supported
    - Much larger ecosystem of solutions and tools
Spark complements Hadoop

- Spark leverages Hadoop ecosystem
  - HDFS, HCatalog, Data Input/OutputFormats
  - Huge investment in data collection & tooling
Spark the “lingua franca”

- Data scientists & Developers need an open standard for sharing their Algorithms & functions, an “R” for big data.

- Spark best current candidate:
  - Open Source - Apache Foundation
  - Expressive (MR, iteration, Graphs, SQL, streaming)
  - Easily extended & embedded (DSLs, Java, Python…).
**Trend 2:** most operational DBMSs will offer multiple data models, relational and NoSQL, in a single DBMS platform.
Operational databases

- Operational database management systems are used to manage dynamic data in real-time.

- Operational databases use NoSQL DBMS engines and distributed database architecture that provides high availability and fault tolerance.
Gartner Magic quadrant for operational database management systems

- By 2017, all leading operational DBMSs will offer multiple data models, relational and NoSQL, in a single DBMS platform.
Using SQL to query NoSQL and relational databases

- SQL: SELECT * FROM NoSQL WHERE category='NoSQL' (Support by CouchBase)

- JSON results:

```json
{
  "name": "Couchbase Server",
  "version": "4.0",
  "category": "NoSQL",
  "features": [
    "name": "N1QL",
    "capabilities": ["JOIN", "NEST", "UNNEST"]
  ]
}
```
Using SQL to query NoSQL and relational databases

- SQL: SELECT * FROM RDB WHERE category='RDB'

- Relational results:

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>category</th>
<th>Features_name</th>
<th>Features_capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>4.0</td>
<td>RDB</td>
<td>SQL</td>
<td>JOIN, NEST, GROUP-BY</td>
</tr>
</tbody>
</table>
Using SQL to join NoSQL and relational databases

- SQL: SELECT * FROM RDB, NoSQL WHERE RDB.name = NoSQL.name

- Relational results:

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>category</th>
<th>name</th>
<th>capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE</td>
<td>1.0</td>
<td>NoSQL</td>
<td>SQL</td>
<td>JOIN</td>
</tr>
</tbody>
</table>

- JSON results:

```json
{
  "name": "ORACLE",
  "version": "12c",
  "category": "RDB",
  "features": [
    "name": "SQL",
    "capabilities": ["JOIN"]
  ]
}
```
**Trend 3:** Self-service data preparation tools are exploding
Self-service data preparation

- Business users want to reduce the time and complexity of preparing for analysis big data

- Products: Alteryx, Trifacta, Paxata
Alteryx

• Alteryx is an American computer software company based out of Irvine, California.

• The company's products are used for data blending and advanced data analytics.

• Alteryx has a stated goal of enabling advanced analytics to be performed by non-specialists.
Alteryx workflow
Trifacta

- Trifacta is a data transformation platform provider that enables business analysts, data scientists and IT programmers to transform data into a usable form for analysis.
• Paxata develops self-service data preparation software

• Paxata's software is used to combine data from different sources, then check it for data quality issues, such as duplicates and outliers.

• Algorithms and machine learning automate certain aspects of data
Three trends for big data management in 2016:

1. Spark grows fast and more popular
2. One DBMS will host NoSQL and SQL
3. Popularity of self-service data preparation tools will explode.