Towards Modeling Approach
Enabling Efficient Platform for
Heterogeneous Big Data Analysis

Andrey.Sadovykh@softeam.fr
Outlines

- Introduction
- Model-driven development
- Big Data
- Juniper
- Case-study results
- Conclusions
SOFTEAM – a French IT services / Software vendor

- SOFTEAM, a growing company
  - 25 years’ experience
  - 850 experts
  - Regular growth

- Specialist in OO technologies, new architectures, methodologies

- Banking, Defense, Telecom, ...

2005: 17.5 ME  
2006: 20 ME  
2008: 23 ME  
2013: 70 ME

www.modeliosoft.com
Modelio for Software and System Engineering

- UML editor with 20 years’ history
  - CloudML
  - SysML
  - MARTE
  - Code generation
  - Documentation
  - Teamwork

- Available under open source at Modelio.org
MODEL-DRIVEN DEVELOPMENT
It is all about models ... Starting with UML

Requirements

UML Use Cases

Architecture

UML Components and Classes

Design

Revised Classes or Domain Specific Language

Implementation

Code generation

Java, C++, Frameworks
Model = Code
Typical example: Control system for a frigate

- 800+ components
- Developed by 100+ engineers
- 1M+ LOC

MDD fosters Productivity and Quality with
  - Code generation
  - Components reuse
  - Tracing
  - Automation
Curious DSL example: Ruby on Rails

### HAML

<table>
<thead>
<tr>
<th>Haml</th>
<th>HTML</th>
</tr>
</thead>
<tbody>
<tr>
<td>%br{:clear =&gt; 'left'}</td>
<td>&lt;br clear=&quot;left&quot;/&gt;</td>
</tr>
<tr>
<td>%p.foo Hello</td>
<td>&lt;p class=&quot;foo&quot;&gt;Hello&lt;/p&gt;</td>
</tr>
<tr>
<td>%p#foo Hello</td>
<td>&lt;p id=&quot;foo&quot;&gt;Hello&lt;/p&gt;</td>
</tr>
<tr>
<td>.foo</td>
<td>&lt;div class=&quot;foo&quot;&gt;...&lt;/div&gt;</td>
</tr>
<tr>
<td>#foo.bar</td>
<td>&lt;div id=&quot;foo&quot; class=&quot;bar&quot;&gt;...&quot; &lt;/div&gt;</td>
</tr>
</tbody>
</table>

Feature: User can manually add movie

Scenario: Add a movie

Given I am on the RottenPotatoes home page
When I follow "Add new movie"
Then I should be on the Create New Movie page
When I fill in "Title" with "Men In Black"
And I should see "Men In Black"
What do we get from MDD?

Pros

• Design **once**, deploy **everywhere**!
• Write your transformation **once**, transform **anything**!

Cons

• Transformations are **hard** to write...
• How to **make sure** they are CORRECT? i.e.
  – Is there any data/semantic loss?
Volume, variety, velocity

1. @-mails sent every second: 2,9 million
2. Video uploaded to YouTube every minute: 25 hours
3. Data processed by Google every day: 24 petabytes
4. Tweets per day: 50 million
5. Products ordered on Amazon per second: 73 items
Only 0,5 % of data is analyzed

- In **2012**, 2 837EB generated - just 0,5% actually analyzed.

  That still amounts to **14EB** (or 14.185 million terabytes)

Source: IDC & EMC
Do you have some data and a problem to solve?

Data fits in memory?

Data fits on single RAID array?

Tons of options. Don’t need database or Hadoop

Solvable with SQL?

Can you program?

Use MySQL

Write a prog.

Dead end
SQL or Hadoop (continued)

Data fits on single RAID array?
- no
  - Have lots of money?
    - no
    - Solvable with Oracle SQL?
      - no
      - Solvable using MapReduce?
        - no
        - Can you program MapReduce jobs?
          - no
          - Write MapReduce on Hadoop?
            - no
            - Dead end
          - yes
        - yes
      - yes
    - yes
  - yes
- no
  - Roll your own MPI solution
- yes
  - Do you have a PhD in parallel prog.?
    - no
    - Solvable using MapReduce?
      - yes
      - Can you program MapReduce jobs?
        - yes
        - Write MapReduce on Hadoop?
          - yes
          - Roll your own MPI solution
        - no
        - Dead end
      - no
    - yes
  - yes

*Inspired by: Aaron Cordova*
Challenges

Hadoop MapReduce is the major trend

Success relies on personnel programming skills

Many problems are not solvable with Hadoop. Real-time?

MPI for high performance computing is an option when you have a lots of money and a PhD
JUNIPER integrates Big Data technologies over MPI.
Modelling in Juniper

- **Models**
  - High level Architecture (Nodes, Programs, Streams…)
  - Real-time constraints

- **Reverse Engineering**

- **Code Generation** (+MPI initialization, communication, etc)

- **Deployment Scripts** (in progress)

- **Code Generation**

- **Measurements & Advice**

- **Configuration**

- **Model Export**

- **Schedulability Analysis Tool** (in progress)

- **Scheduling Advisor** (in progress)
Mapping Programming Model, UML and MARTE

Programming Model
- JUNIPER Program
- Channel
- Cloud Node

UML
- Class
- Instance:

MARTE
- <<RtUnit>>
- Class
- <<HwComputingResource>>
- Instance:
Modelling the application and real-time constraints

- Real-time constrains
  - response time
  - bandwidth

JUNIPER Programs

Big Data flow

ProgramA

communication

ProgramB

specification
- relDI=(100,ms)
- occKind=(period=12ms)

<<SwMutualExclusionResource>>

Lock
Modelling the hardware infrastructure at a high level

Cloud Node

juniperCloudNode1:

<<HwProcessor>>
cpu:
{HwProcessor.nbCores(4)}

<<HwDrive>>
disk:

CPU with 4 cores

Hard drive
MPI code generation

JUNIPER Application Model

Code Generation

```java
public class ProgramA {
    private static final Logger logger = Logger.getLogger(ProgramA.class);

    public static final int RANK = 0;

    public static final int MESSAGE_TAG = 0;

    public static void sendMessageToProgramB(int msg) throws Exception {
        MPI.COMM_WORLD.send(new int[]{msg}, 1, MPI.INT, ProgramB.RANK, MESSAGE_TAG);
    }
}
```
PETAFAUEL CASE STUDY
Prepaid Credit Card Fraud Makes Criminals Millionaires

May 29, 2013 by Paul McCormack

1 comment(s)

**Total Loss: $45 Million**

By now, I’m sure you’ve seen the headlines and read the details about the multi-million dollar global cyberheist. Yet again, cybercriminals have landed a huge payday. In a little more than half a day’s time collectively, thieves stole approximately $45 million via prepaid credit cards. As many ask, “How has this happened again?”, I’ll take a moment to break it down.

In this case, cybercriminals hacked the databases of one, possibly two payment processors (details are still unfolding). Since the credit limits on prepaid cards are far lower than the rates on traditional cards, the fraudsters inflated the available balances and removed the daily withdrawal limits. They then sent the card data and corresponding personal identification numbers (PINs) to their “cashers” around the world to encode on the plastic cards. The cashers, located in 24 countries, rushed to their nearest ATMs and withdrew cash – lots of it. All the while, the cybercriminals stay connected to the third-party processors networks and watched the withdrawals taking place in real time (they have checks and balances in place to ensure that the cashers don’t get too greedy). The final step involved laundering the cash via the purchase of large ticket items, including two Rolex watches, a Mercedes SUV, and a Porsche.
Master Card debit card approval within 4 sec

petaFuel - transaction approval within 4 seconds

Transactions
Events
Stream

Historical Data
30 GB

Basic Checks

Fraud Patterns Detection

Transaction Approval Decision
Deployment model

- <<CloudNode>> <<HwComputingResource>>
  172.18.2.81:
  
  <<ProgramInstance>>
  : QueryProcessor
  
  <<ProgramInstance>>
  : PrepaidDB
  
  <<ProgramInstance>>
  : QueryInterface
  
  <<ProgramInstance>>
  : EventProcessor

- <<CloudNode>> <<HwComputingResource>>
  172.18.2.82:

  <<ProgramInstance>>
  : KeyValue Store

  <<CloudDisk>> <<HwDrive>>
  Disk:

  <<CloudNodeCPU>> <<HwProcessor>>
  CPU:
public class EventProcessor {
    public static final int RANK = 1;
    public static IEventProcessor iEventProcessorImpl = new IEventProcessor() {
        @Override
        public void process(Event event) {
            String key = getKeyFromTimestamp(event.getTimestamp());
            String value = keyValueStoreIKeyValueStore.find(key);
            if (value == null) {
                keyValueStoreIKeyValueStore.put(key, "1");
            } else {
                int count = Integer.parseInt(value);
                keyValueStoreIKeyValueStore.put(key, ""+(count+1);
            }
        }
    };
    ...
    public static void main(final String[] args) {
        MPI.Init(args);
        ...
        MPI.Finalize();
    }
}
CONCLUSIONS
## Juniper trade-offs

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Hadoop</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>HDFS - file system (httpd)</td>
<td>HPC cluster interconnect (Infiniband)</td>
</tr>
<tr>
<td>Data flow</td>
<td>Map Reduce</td>
<td>Modeling + MPI comms</td>
</tr>
<tr>
<td>Parallelization</td>
<td>Automatic</td>
<td>Manually based on domain decomposition</td>
</tr>
<tr>
<td>Response time guaranties</td>
<td>None</td>
<td>Real-time for single node</td>
</tr>
<tr>
<td>Stages in multi-format</td>
<td>No</td>
<td>Any (incl. Hadoop + FPGA)</td>
</tr>
<tr>
<td>Hardware</td>
<td>Commodity cluster</td>
<td>HP cluster</td>
</tr>
<tr>
<td>Price</td>
<td>€</td>
<td>€€€</td>
</tr>
<tr>
<td>Skills</td>
<td>+</td>
<td>+++</td>
</tr>
<tr>
<td>Customers</td>
<td>General audience</td>
<td>Critical systems</td>
</tr>
</tbody>
</table>
Prospects – more work

Work in progress

- UML Based Language
  - MPI Communication
  - Timing properties
  - Deployment

- petaFuel case study

Future work

- Modelling payload
- Integrating schedulability
- Running final evaluations
- Final release
Questions?

Andrey Sadovykh
Marcos Almeida
SOFTEAM | ModelioSoft
{name.surname}@softeam.fr

SOFTEAM R&D Web Site:
http://rd.softeam.com

Modelio Web Site:
http://www.modelio.org

JUNIPER Web Site:
http://www.juniper-project.org

*for your questions