# Dell Software Group "IT System Management Introduction” Lab Course at Sun Yat-sen University

Lecturers: Beam Liu, Kyle Wang, Lewis Liu, Russell Chen

Date: Sep-12 to Sep-14

Place: Sun Yat-sen University

Contents

[Dell Software Group "IT System Management Introduction” Lab Course at Sun Yat-sen University 1](#_Toc397761026)

[Introduction 2](#_Toc397761027)

[Purpose & Goal 2](#_Toc397761028)

[Required technical skills 2](#_Toc397761029)

[APM Prototype Content 2](#_Toc397761030)

[Requirement 2](#_Toc397761031)

[Use cases 2](#_Toc397761032)

[Architecture diagram 3](#_Toc397761033)

[UI Prototype 4](#_Toc397761034)

[Used technologies 4](#_Toc397761035)

[Lectures arrangement 4](#_Toc397761036)

[Libraries and technologies reference 5](#_Toc397761037)

[Javascript Instrumentation 5](#_Toc397761038)

[Java Instrumentation 5](#_Toc397761039)

[User interface 5](#_Toc397761040)

[Environment preparation 6](#_Toc397761041)

[Bonus task 6](#_Toc397761042)

# Introduction

## Purpose & Goal

The purpose of this lab course is to present a prototype of application performance monitoring (APM), introduce the latest and advanced technologies in APM, in this lab course, with the guidance of lectures, the students will try to build a small prototype of APM product. Through this lab course, the students will learn how to use all kinds of advanced technologies to implement a simple performance monitoring product and have an overview of the APM technology trend.

## Required technical skills

* Familiar with Java
* Familiar with JS and HTML

# APM Prototype Content

## Requirement

There is a website running in internet, however, the site administrator received some complains about the performance, sometimes, the website runs very slow, however, the site administrator has no concept of that, he/she doesn’t know how to begin, he/she need a great product that can help to dig out the root cause of the performance issue. The performance issue should be resolved before customers’ complains coming.

Here are some additional information of the website:

* The website is built in Java language
* Tomcat is the app container
* Used a database to store data

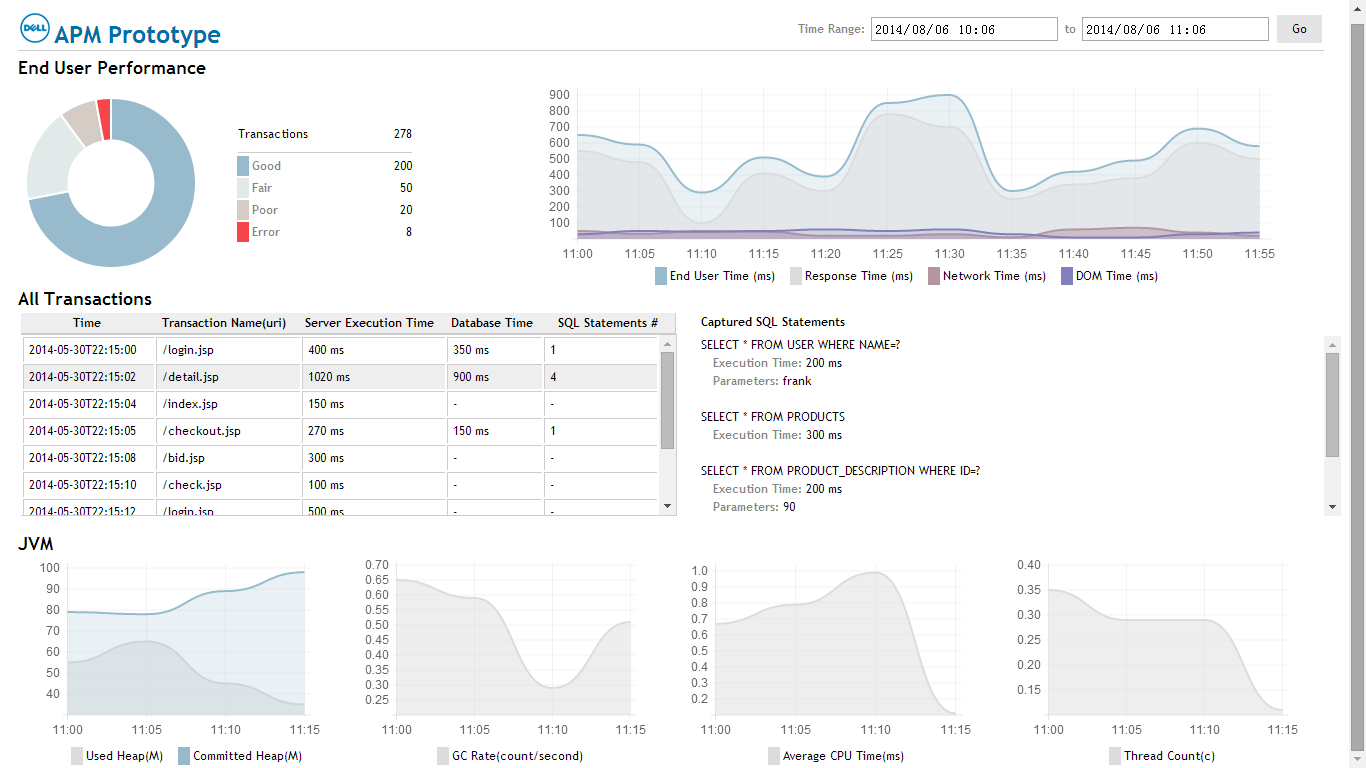
## Use cases

* As a website administrator, I want to know exactly, how many transactions (requests) my website is handling during a certain time ago till now, what is the real end user performance. At the same time, it is better that I could know the health of those transactions, are they fast enough? I want to know the overall state, and then I could take some actions before receiving users’ complains.
* In case I found some performance issue, it will be great if I can know which part consumes most of the time at server side, is it possible that the tomcat handles requests very slow? Or the most possibility of database query doesn’t hit an index? It will be even greater, if I can get all the SQL statements and their execution time. I will find corresponding partners to resolve those issues, after I get such kind of information.
* As my website is running in tomcat, I would like to know some critical information of the JVM, such as the JVM heap usages, garbage collection, cpu consuming and threads.

## Architecture diagram



## UI Prototype



## Used technologies

In this prototype, it uses the modern [instrumentation](http://en.wikipedia.org/wiki/Instrumentation_(computer_programming)) technologies to get all kinds of performance counters. Use javascript instrumentation to retrieve client side performance data, such as DNS lookup time, connection time, document receiving time, DOM rendering time and so on; use java instrumentation to retrieve server side performance data, such as JVM related performance data, database call time and so on. For the user interface building, beside some standard html, js and css technologies, it also uses Chart.js to build all kinds of charts.

# Lectures arrangement

Day one morning

* Lab course introduction
* APM technology introduction and end user performance monitoring
* Architecture & user interface of APM prototype

Day one afternoon

* Create APM prototype object model
* Instrumentation implementation of application server

Day two morning

* Instrumentation implementation of application server -- continue

Day two afternoon

* JS instrumentation
* Link end user data with application server data

Day three morning

* Build UI to view APM data
* Package the APM prototype

Day three afternoon

* Open discussion of what can be better in our APM prototype
* Introduction of our real APM SaaS product
* Q&A

# Libraries and technologies reference

## Javascript Navigation Timing

* Navigation Timing -- <http://www.w3.org/TR/navigation-timing/>
* XMLHttpRequest -- <http://www.w3.org/TR/XMLHttpRequest/>
* Measuring the speed of resource loading with JavaScript -- <http://blog.trasatti.it/2012/12/measuring-the-speed-of-resource-loading-with-javascript-and-html5.html>

## Java Instrumentation

* JMX & MXBeans -- <http://docs.oracle.com/javase/tutorial/jmx/index.html>
* Java Instrumentation

<http://docs.oracle.com/javase/7/docs/technotes/guides/instrumentation/index.html>

<http://www.ibm.com/developerworks/cn/java/j-lo-jse61/>

<http://blog.csdn.net/pwlazy/article/details/510974>

* Javassist

<http://www.csg.ci.i.u-tokyo.ac.jp/~chiba/javassist/>

<http://blog.csdn.net/yadandan520_ya/article/details/3956867>

* HttpClient

<http://hc.apache.org/httpclient-3.x/tutorial.html>

## User interface

* HTML5 -- <http://www.w3school.com.cn/html5/>
* Chart.js -- <http://www.chartjs.org/>
* Jquery -- <http://jquery.com/>

## Ant build

<http://ant.apache.org/>

# Environment preparation

* Oracle Java SDK 1.7, please make sure java environment is ready
* Tomcat 7.0
* Latest Chrome
* Standard Eclipse 3.7+
* An advanced text editor tool, such as EditPlus, EditPlus++, UtraEdit
* MySQL database
* Install a target application [zencherrycms](http://sourceforge.net/projects/zencherrycms/) using tomcat and MySQL, make sure it could run locally

# Bonus task

* A document about the improvement of APM prototype.
* A document about how to do performance monitoring in any other programming language, such as PHP, .Net, Ruby, Nodejs and so on. It could be the container (app server) monitor or request tracing monitoring, even better, write down both of them. Some pseudo codes are welcome.