Task Process Monitor

Objective:

To monitor the real time process of a machine that includes but not limited to usage of cpu, memory, network, etc.

Trioxes Team
Vivek Vellaiyappan Surulimuthu
Deeksha Sudini
Neha Ersavadla

AIM:

- To show informative statistics about the computer's performance.
- To have a real time information on memory leaks, concurrency issues
- To achieve the information of System level,CPU,Process level,File system and Network interface level statistics in the UI

Approach

Following SDLC process

- Initiation of the idea of task process monitor based on the specifications given
- Developing the concept of task process monitor
- Project Management: Agile methodology (Scrum)
- Software Product Development Path: (following MVPs)
- Deciding the tech stack
- Gathering system level information by parsing the information to get better data insights and showing the info on the website page.
- Approaching towards the project in MVC model where the controller servlet fetches the data from the model to the view.
- Fetching real time data and retrieving the data to the view part.
- Showing the process timings and graphs related to the memory usage and time.
- Cleaning the unnecessary usage of resources to improve the performance.

How we started?

Meeting Logs:
https://docs.google.com/document/d/16KIo6blGBbdbOINyLrJPagyat-QKd8JMok9Hglf-IRU/edit#

Work Logs:
https://docs.google.com/document/d/1i1reEmOdP9Uy94XcugNsKHaqIT_aHs8EsjHF8UMoeI4/edit#

Project-Hub:
https://docs.google.com/document/d/1-87xMxe7MHWXLTlZeG8o35D9Z7L6nnqi8huZe12Vsq4/edit#heading=h.rp4j3q1bp3y8

Research Notes:
https://docs.google.com/document/d/1UGXD3kV3vcRh1YUrGE9c1tHMExFZGaEgZRzmDbDGnMM/edit

Roles Delegation

Name	Roles	Shared Roles
Vivek Vellaiyappan	Scrum Master, Manager, Software Engineer, Web Developer, DevOps Engineer	Architect Technical Writer
Deeksha	Testing Engineer Automation Engineer	
Neha		

Roles & Responsibilities - Vivek

Scrum Master

- Scrum Methodology
  - Stand Ups, Retrospective, Project Sprints
- Meeting Log
  - - contains all our meeting notes
- Work Log
  - Individuals log every work they did for Project

Manager

- Removing road blocks for project

Software Engineer

- Learnt Spring Boot Technology & Implemented all the backend works for the project
- Followed Best Coding Practices

Web Developer

- Handled the front end UI

DevOps Engineer

- Integrated CI - Semaphore
- Dockerized the application
- Heroku deployment (in progress)

Roles & Responsibilities - Deeksha & Neha

Deeksha - Tester / Technical Writer

Performed testing on the application and raised bugs in the github.

Automation Engineer - Written automation code to test the application

Technical Writer - Worked on documentation and presentations. Researched about the technologies and presented KT to the team mates.

Responsible for maintaining the meeting logs and work logs.

Neha - Tester / Technical Writer

Performed testing on the application and raised bugs in the github.

Automation Engineer - Written automation code to test the application

Technical Writer - Worked on documentation and presentations. Researched about the technologies and presented KT to the team mates.

Responsible for maintaining the meeting logs and work logs.

Tech Stack

- Java
- Sigar
- Spring Boot, Spring,
- Spring MVC
- Spring Rest Controller
- Spring Data JPA
- Hibernate
- Spring Scheduler
- H2 - Database
- Junit
- HTML, CSS, JavaScript, jQuery, Bootstrap
- Github
- CI/CD pipeline - Semaphore
- Docker
- Maven
- Swagger
- Webjarjs
- Log4j

Project Code Based Statistics

As of now(12/6/2018) the code based statistics are:

Releases: 3

Branches: Total 43

Merged 36

Unmerged 7

Commits: 66

Pull requests: 16 Active

16 Merged

Issue tracker

- Total issues created 103
- Total closed 65
- Total open still 38

Labelled Issues (some are combined)

- Enhancement 47
- Bug 30
- Testing 17

Project Sprints: 4

Maven Build

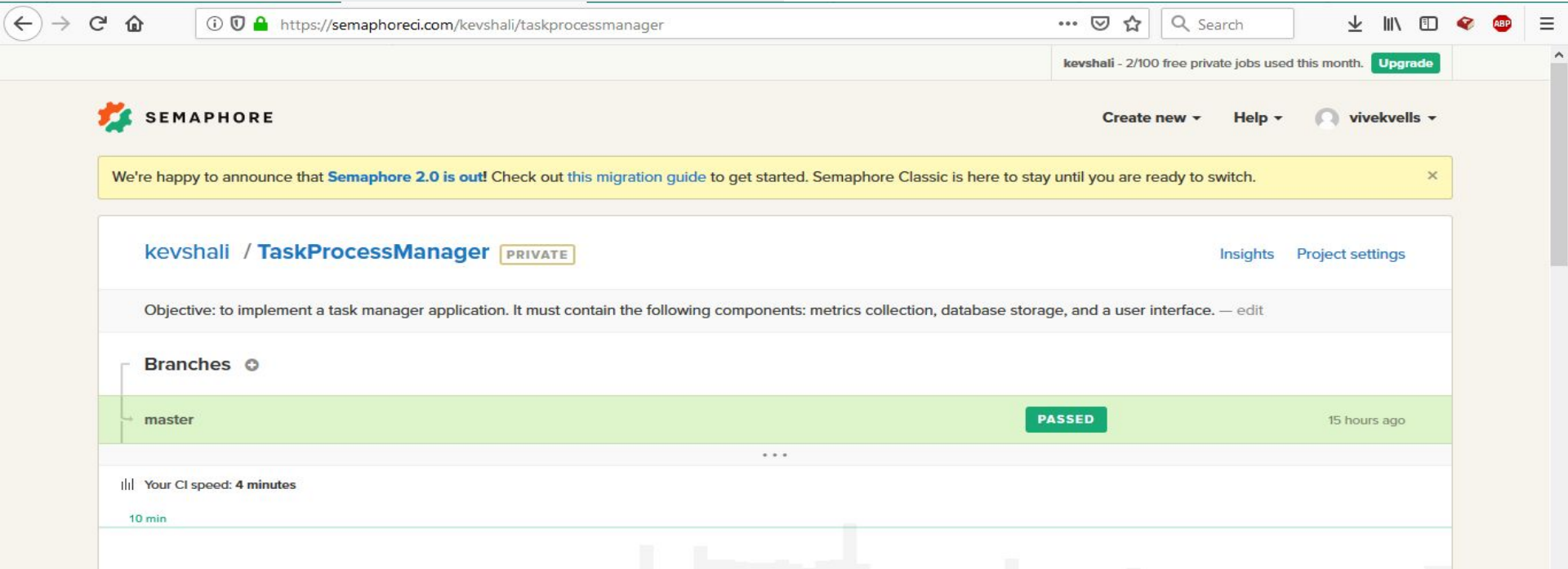
```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 51, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ processmonitor ---
[INFO] Packaging webapp
[INFO] Assembling webapp [processmonitor] in [E:\kevDev\ProjectWorks\TaskProcessManager\version1\processmonitor\target\processmonitor-0.0.2-SNAPSHOT]
[INFO] Processing war project
[INFO] Webapp assembled in [2018 msecs]
[INFO] Building war: E:\kevDev\ProjectWorks\TaskProcessManager\version1\processmonitor\target\processmonitor-0.0.2-SNAPSHOT.war
[INFO]
[INFO] --- spring-boot-maven-plugin:2.1.0.RELEASE:repackage (repackage) @ processmonitor ---
[INFO] Replacing main artifact E:\kevDev\ProjectWorks\TaskProcessManager\version1\processmonitor\target\processmonitor-0.0.2-SNAPSHOT.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 01:10 min
[INFO] Finished at: 2018-12-06T18:06:32-05:00
[INFO] Final Memory: 47M/357M
[INFO] ------------------------------------------------------------------------

E:\kevDev\ProjectWorks\TaskProcessManager\version1\processmonitor>java -jar target\processmonitor-0.0.2-SNAPSHOT.war
```

Semaphore CI

Task Process Manager / ProcessOnitor / ProcesSanager

BUILD STATUS PASSED



Dockerized Application

```
Vivek-Pc@kev MINGW64 /e/kevDev/ProjectWorks/TaskProcessManager/version1/processmonitor (DockerizeApplication)
$ mvn clean package docker:build
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building processmonitor 2.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ processmonitor ---
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:resources (default-resources) @ processmonitor ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 70 resources
[INFO] Using authentication suppliers: [ConfigFileRegistryAuthSupplier]
[INFO] Copying E:\kevDev\ProjectWorks\TaskProcessManager\version1\processmonitor\target\processmonitor-2.0-SNAPSHOT.war -> E:\kevDev\ProjectWorks\TaskProcessManager\version1\processmonitor\target\docker\processmonitor-2.0-SNAPSHOT.war
[INFO] Building image processmonitor
Step 1/3 : FROM java:8

 ---> d23bdf5b1b1b
Step 2/3 : ADD /processmonitor-2.0-SNAPSHOT.war //

 ---> a130f2a109f2
Step 3/3 : ENTRYPOINT ["java", "-jar", "/processmonitor-2.0-SNAPSHOT.war"]

 ---> Running in ad97c14b102b
Removing intermediate container ad97c14b102b
 ---> 5e102b1b96d3
ProgressMessage{id=null, status=null, stream=null, error=null, progress=null, progressDetail=null}
Successfully built 5e102b1b96d3
Successfully tagged processmonitor:latest
[INFO] Built processmonitor
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 01:35 min
[INFO] Finished at: 2018-12-06T18:19:26-05:00
[INFO] Final Memory: 53M/506M
[INFO] ------------------------------------------------------------------------

Vivek-Pc@kev MINGW64 /e/kevDev/ProjectWorks/TaskProcessManager/version1/processmonitor (DockerizeApplication)
$
```

Highlights

- SDLC process experienced
- Architectural Design
- Agile - Scrum
- GitHub SCM & GitHub issues as Bug Tracker
- Maven Build System
- CI: Semaphore
- CD: Docker container
- Deployment: Heroku (In progress)
- REST API Endpoints
- API Documentation: Swagger API

Future plans/ Enhancements

- Cloud Deployment
  - Scaling up & delivery of this web Application to everyone by using Cloud CSPs
- Going Microservices
- Developer Portal for exposing application APIs
- Code Development Related
  - Hot Deploy concept in concept tryout
  - Issues creation automatically when build fails
- Using Front-end framework
- Application Analysis

Demo:

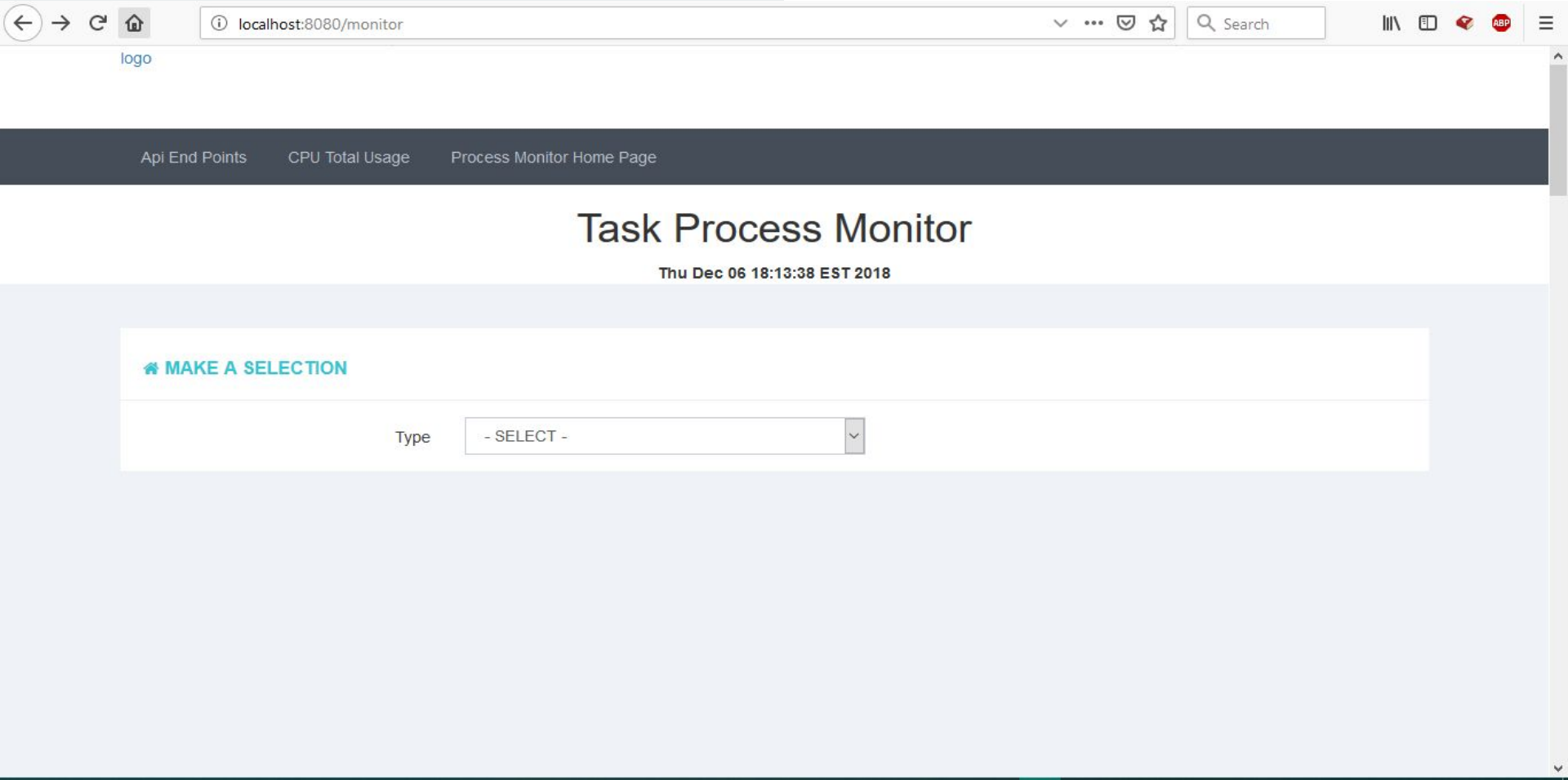
Run the application on localhost

http://localhost:8080/endpoint

https://github.com/vivekVells/TaskProcessManager/issues

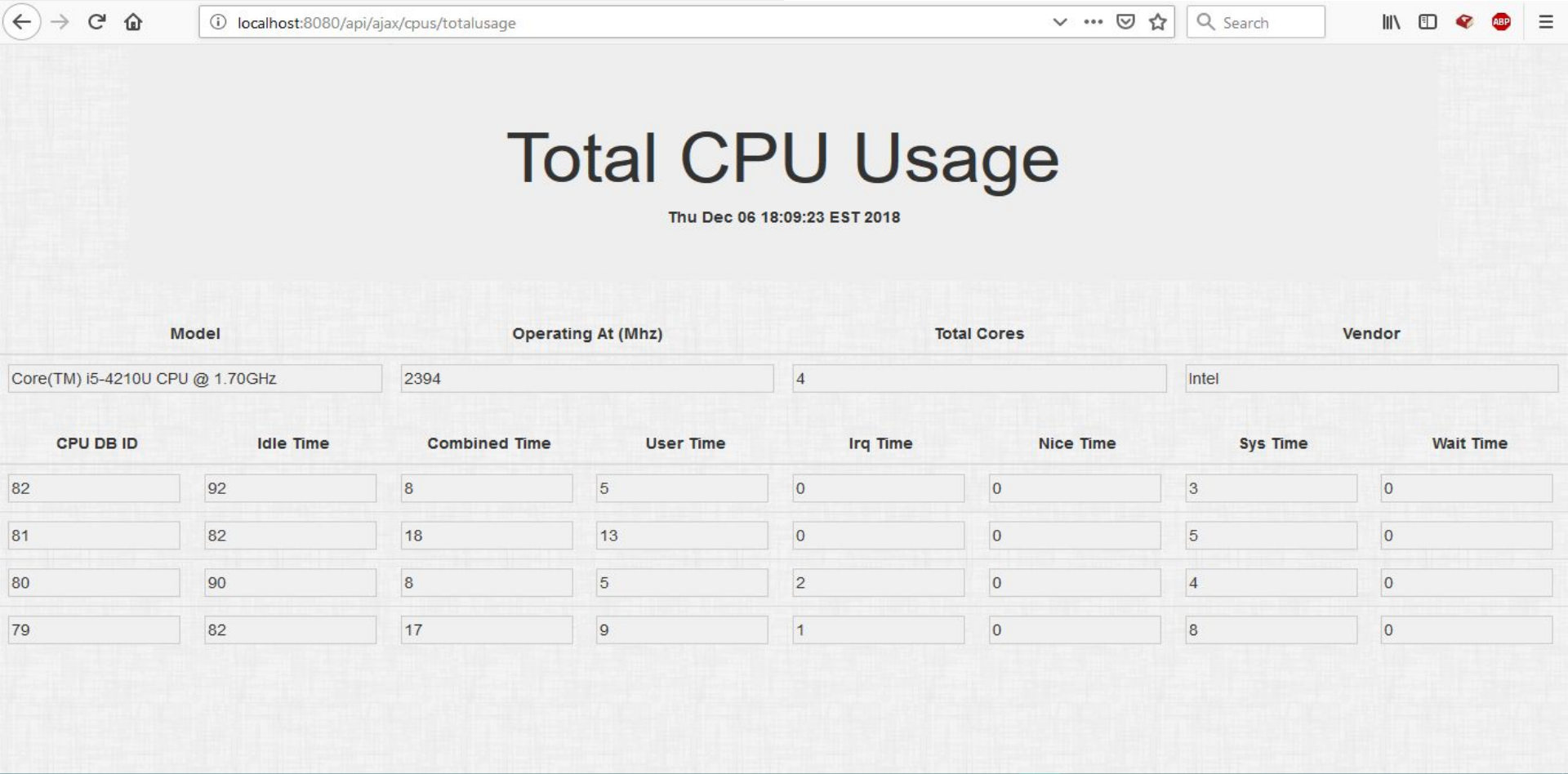
Demo:

Total CPU Mem Usage



Demo:

Total CPU Mem Usage



References:

- https://www.javatips.net/api/oshi-master/oshi-core/src/main/java/oshi/SystemInfo.java
- https://www.programcreek.com/java-api-examples/?api=oshi.util.FormatUtil
- https://code.dblock.org/2010/06/23/introducing-oshi-operating-system-and-hardware-information-java.html
- https://github.com/java-native-access/jna
- https://github.com/oshi/oshi/blob/master/oshi-core/src/test/java/oshi/SystemInfoTest.java
- http://download2.nust.na/pub4/sourceforge/s/si/sigar/docs/sigar1.pdf

Thank You