

ZPR Projekt: Monitor Zasobów – 2

Szkielet Aplikacji – Krótka dokumentacja

1. Prosty Diagram UML Projektu:



2. System Monitor:

System Monitor jest to aplikacja, która zajmuje się zbieraniem użytecznych informacji nt. systemu na którym się znajduje. Informacje te wysyła do bazy danych. Podzielona jest na moduły (nazwy są abstrakcyjne i nie mają jeszcze przełożenia na istniejący do tej pory kod):

- Sensor Information Provider – moduł logiczny, który zajmuje się zbieraniem informacji z podzespołów komputera.
- List of active processes Provider – przechowuje listę aktywnych procesów z informacjami na ich temat

3. CassandraDB:

Baza danych oparta na systemie zarządzania Apache Cassandra. Przechowuje informacje uzyskane od aplikacji System Monitor i udostępnia wybrane dla wybranego serwera http.

4. HTTP Server:

Server http, w jeszcze nie wybranej przez nas technologii (prawdopodobnie Django). Posiada dwa moduły abstrakcyjne moduły:

System Information Selector – moduł wybierający z bazy danych informacje użyteczne dla użytkownika

Chart Maker – moduł tworzący wykresy z zebranych danych

Threshold Notifier – sprawdza informacje i weryfikuje czy użytkownik musi zostać powiadomiony.

5. Komunikacja:

Komunikacja bazy danych z aplikacją System Monitor odbywa się za pomocą sterownika DataStax C/C++. Baza danych i aplikacja mogą działać na jednej fizycznej maszynie, ale nie muszą. Komunikacja bazy danych z serwerem prawdopodobnie będzie się też odbywać, za pomocą sterownika DataStax (najprawdopodobniej python driver), ale na tym etapie jeszcze nie możemy powiedzieć jakiego. Wstępnie zakładamy, że server i maszyna z uruchomioną aplikacją leżą w tej samej podsieci.

6. Co zostało zrobione do tej pory:

Został napisany program realizujący po części założenia aplikacji System Monitor. Do tego, został napisany build system w CMake, aby ujednolicić proces budowania na systemie operacyjnym Windows oraz Linux. Struktura programu jednak bardzo się różni dla tych systemów. Spowodowane jest to zasadniczą różnicą w uzyskiwaniu tych samych informacji. Dla Windows'a została użyta biblioteka OpenHardwareMonitorLib napisana w C#. Do użycia jej, potrzebne było zastosowanie język C++/CLI, którego nie da się wykorzystać w Linuxie. Informacje w Linuxie zaś, odczytywane są z odpowiednich plików.

7. Co zostało do zrobienia:

- a. Dokończyć implementację aplikacji System Monitor
- b. Implementacja serwera http
- c. Usprawnić testowanie
- d. Usprawnić budowanie

8. Budowanie programu – do budowania wykorzystywany jest CMake. Aby przeprowadzić unit testy dla Windowsa, trzeba zbudować za pomocą komendy `cmake <source_dir> <build_dir> -DCMAKE_BUILD_TYPE="Debug"` i wywołać program w Visual Studio w trybie Debug – inaczej unit testy nie zadziałają. Jest to prawdopodobnie spowodowane bugiem w kompilatorze MSVC.