Politechnika Warszawska, WEiTI

pRoman - process manager

Projekt TIN. Dokumentacja wstępna

Damian Bułak
Michał Sypetkowski
Marcin Waszak
Ahata Valiukevich

20 kwietnia 2017

Treść zadania

W sieci jest zbiór zarządzanych węzłów, serwer zarządzający i stacja konsoli administratora. W węzłach pracują agenty zarządzające. Agent zarządzający może: załadować kod nowego procesu, usunąć kod procesu, uruchomić/zatrzymać/wznowić/zabić dany proces zgodnie z harmonogramem, wznowić proces nie raportujący swej żywotności, podać dane statystyczne serwerowi. System umożliwia administratorowi zarządzanie rozproszonymi procesami. System komunikacji powinien móc pracować w przestrzeni adresów *IPv4* i *IPv6*. Ponadto należy zaprojektować moduł do *Wireshark* umożliwiający wyświetlanie i analizę zdefiniowanych komunikatów.

Przyjęte założenia funkcjonalne i niefunkcjonalne

Założenia funkcjonalne

Implementowany system ma umożliwiać:

- wysyłanie obrazu procesu
- usuwanie obrazu procesu
- wznawianie procesu nie raportującego swej żywotności

- podawanie danych statystycznych serwerowi
- uruchamianie danego procesu zgodnie z harmonogramem
- zatrzymywanie danego procesu zgodnie z harmonogramem
- wznawianie danego procesu zgodnie z harmonogramem
- zabijanie danego procesu zgodnie z harmonogramem

Założenia niefunkcjonalne

- niezawodność komunikacji dzięki użyciu protokołu TCP
- funkcja watchdoga chroniącego przed zawieszaniu się agenta
- intuicyjna obsługa dzięki zastosowaniu poleceń z poziomu konsoli

Podstawowe przypadki użycia

Administrator - użytkownik zalogowany do konsoli administratorskiej

Tablica 1: Podstawowe przypadki użycia

NAZWA	AKTORZY	OPIS
Wysłanie obrazu procesu	Administrator	1. Administrator wybiera proces do załadowania. 2. Proces zostaje załadowany do systemu.
Przeglądanie listy procesów	Administrator	1. Administratorowi prezentowana jest lista załadowanych procesów. 2. Obok każdego procesu administrator może zobaczyć stan procesu (uruchomiony bądź nie), jak i podstawowe informacje o procesie.

Kontynuacja na następnej stronie

Tablica 1 – *Kontynuacja z poprzedniej strony*

NAZWA	AKTORZY	OPIS
Usuwanie obrazu procesu	Administrator	1. [Przeglądanie listy procesów] 2. Administrator usuwa wybrany z listy proces poleceniem *'Usuń'*.
Uruchomienie procesu	Administrator	1. [Przeglądanie listy procesów] 2. Administrator wybiera z listy proces i uruchamia go poleceniem *'Uruchom'*. 3. Następuje przetwarzanie procesu (opis w architekturze systemu). Administrator może zobaczyć postęp przetwarzania procesu w postaci procent zakończenia. 4. Po zakończeniu przetwarzania administrator widzi rezultat przetwarzania.
Zatrzymanie uruchomionego procesu	Administrator	1. [Przeglądanie listy procesów] 2. Administrator wybiera z listy uruchomiony proces. 3. Administrator zatrzymuje proces poleceniem *'Zatrzymaj'*.
Planowanie czasu uruchomienia procesu	Administrator	1. [Przeglądanie listy procesów] 2. Administrator edytuje czas wykonania procesu po wpisaniu polecenia *'Edytuj'*.

Kontynuacja na następnej stronie

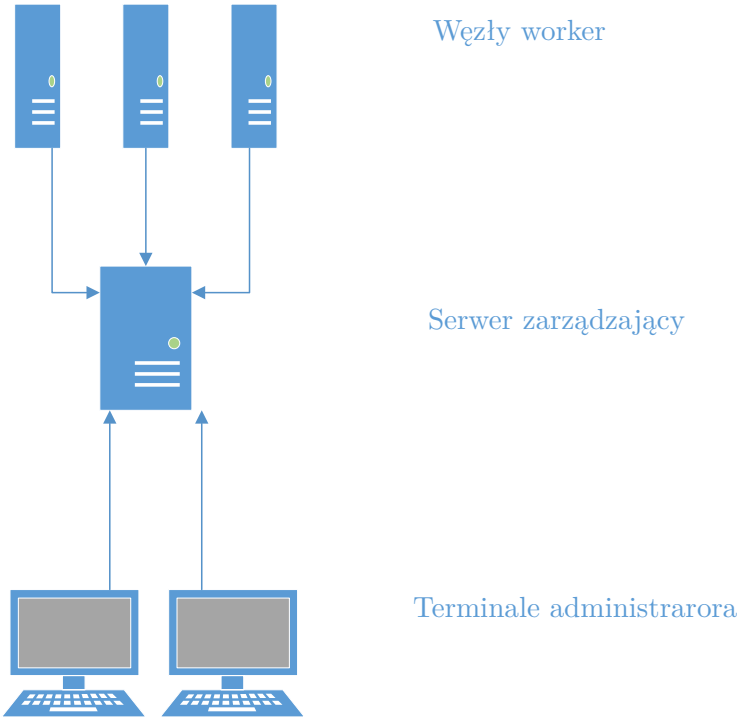
Tablica 1 – *Kontynuacja z poprzedniej strony*

NAZWA	AKTORZY	OPIS
Podgląd harmonogramu procesów	Administrator	1. Administrator przegląda tabelaryczny harmonogram załadowanych procesów.
Podgląd danych statystycznych	Administrator	1. Administrator przegląda dane statystyczne z informacjami o załadowanych, uruchomionych, zatrzymanych procesach oraz o obciążeniu poszczególnych węzłów.

Wybrane środowisko sprzętowo-programowe

1. System operacyjny: ***GNU/Linux***.
2. Język: ***C++***.
3. Biblioteki: STL, boost/filesystem, boost/process, BSD Sockets API.
4. Testowanie: boost/test.
5. Debugowanie: wireshark.
6. Budowanie: cmake

Architektura rozwiązania



Rysunek 1: Ilustracja systemu

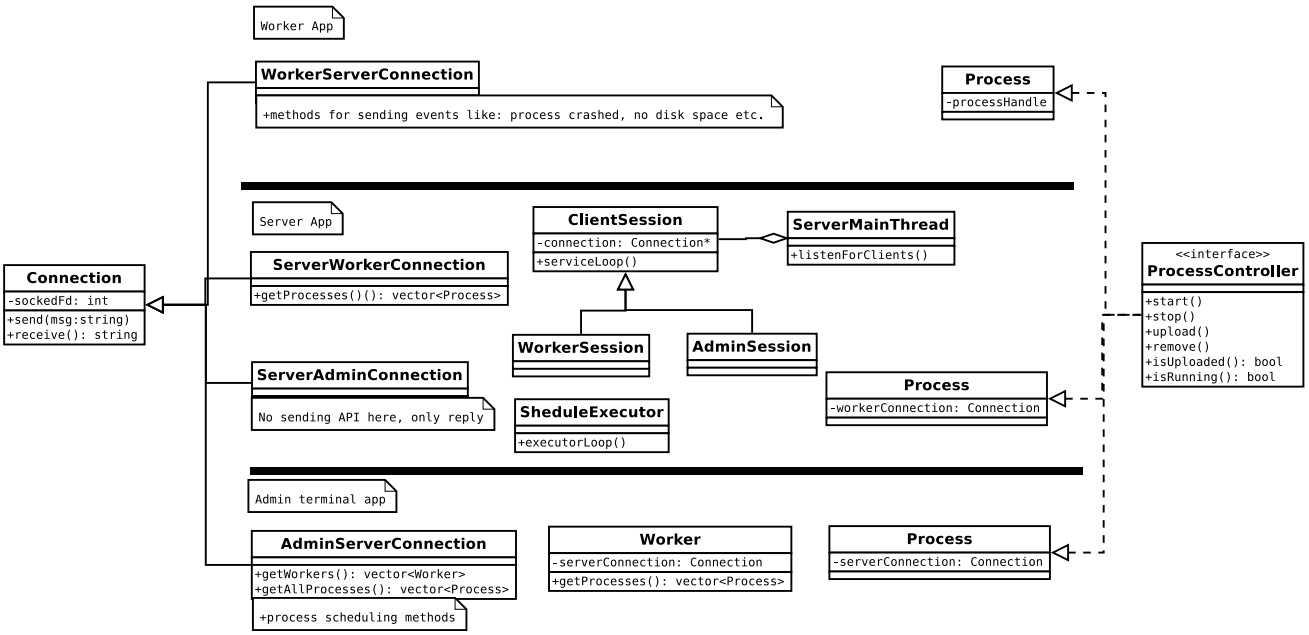
Węzeł worker - maszyna, która wykonuje procesy. Bezpośrednia kontrola procesów jest sprawowana przez procesy sieciowe klasy worker.
Proces klasy worker- specjalny proces sieciowy na maszynach węzłów worker. Jego zadaniem jest zarządzanie procesami na tych maszynach według rozkazów serwera zarządzającego.
Watchdog - specjalny proces obok procesu workera. Stanowi zabezpieczenie przed skutkami zawieszenia procesu workera.
Serwer zarządzający - maszyna, która wysyła rozkazy do agentów maszyn węzłowych. Może przeprowadzać zaplanowane wcześniej czynności. Serwer posiada proces sieciowy klasy server.
Proces klasy server - oczekuje na połączenia od węzłów worker i terminali administratora.
Terminal administratora - maszyna, dzięki której administrator ma możliwość zarządzać procesami na maszynach węzłowych. Każdy taki terminal posiada proces sieciowy klasy administrator.

Planowane aplikacje sieciowe do wykonania:

- aplikacja klasy worker

- aplikacja klasy server
- aplikacja klasy administrator

API modułów stanowiących bloki funkcjonalne



Rysunek 2: Wstępny diagram klas

Sposób testowania

Planujemy użyć testów jednostkowych w poszczególnych aplikacjach naszego projektu, żeby sprawdzić poprawność działania osobnych części całego systemu. Poza testami jednostkowymi napisane będą również testy integracyjne. Przewidujemy rownież testowanie ręczne. Scenariusze testów akceptacyjnych są przedstawione poniżej.

Scenariusze testów akceptacyjnych

Testy akceptacyjne będą polegały na wykonywaniu różnych scenariuszy użycia.

Tablica 2: Przykładowe scenariusze testów akceptacyjnych

CEL TESTU	OPIS PRZEBIEGU	OCZEKIWANY WYNIK
Usuwanie obrazu procesu	Administrator przegląda listę procesów. Wpisuje polecenie do usunięcia procesu oraz jego id.	Jeśli podane id istnieje, obraz procesu zostanie usunięty, w przeciwnym przypadku pojawi się komunikat o braku danego procesu.
Zatrzymanie uruchomionego procesu	Administrator wpisuje polecenie zatrzymania procesu.	Administrator otrzymuje informację o zatrzymaniu procesu.Proces jest widoczny na liście procesów jako załadowany (nie uruchomiony). Jeśli proces nie został jeszcze uruchomiony administrator otrzymuje informację o błędzie.
Załadowanie obrazu procesu	Administrator wpisuje polecenie załadowania nowego obrazu procesu.	Informacja o stanie załadowania procesu - sukces lub nieudane załadowanie, wraz ze wskazaniem przyczyn niepowodzenia w wykonaniu polecenia.
Podgląd harmonogramu procesów	Administrator wpisuje polecenie przeglądu procesów.	Wyświetlona lista procesów wraz z ich planowanymi czasami uruchomienia.
Podgląd danych statystycznych	Administrator wpisuje polecenie służące do podglądu danych statystycznych serwera.	Wyświetlone zostaną informacje o obciążeniu węzłów, liczbie uruchomionych procesów, liczbie załadowanych procesów etc.

Podział prac w zespole

Damian: projekt architektury
Michał: projekt architektury
Agata: projekt architektury, dokumentacja, testowanie, moduł Wireshark
Marcin: team-leader, projekt architektury

Powyższy podział jest zgrubny. Bowiem szczegółowe przydzielanie prac zorganizowane zostało za pomocą systemu tasków na platformie GitHub.

Harmonogram prac

- **19.04** - oddanie dokumentacji wstępnej.
- **30.04** - szkielet programów: serwera, workera oraz administratora; rozpoczęte prace nad skryptami testującymi.
- **8.05** - konsultacje #1: projekt z w pełni działającą komunikacją między administratorem a serwerem (nawiązywanie połączenia, działający mechanizm przesyłania pakietów i poleceń).
- **22.05** - konsultacje #2: działający i prawie skończony projekt, z ewentualnymi błędami niewpływającymi istotnie na działanie programu, rozpoczęcie pracy nad końcową dokumentacją poprojektową.
- **31.05** - oddanie projektu.

Adres projektu na serwerze kontroli wersji

https://github.com/marcin-waszak/DistributedProcesses

W zakadce *Issues* można przeglądać informacje o przydzielanych przez Marcina zadaniach:

- dla kogo zostało przydzielone
- kiedy zostało utworzone
- na jakim etapie realizacji obecnie jest