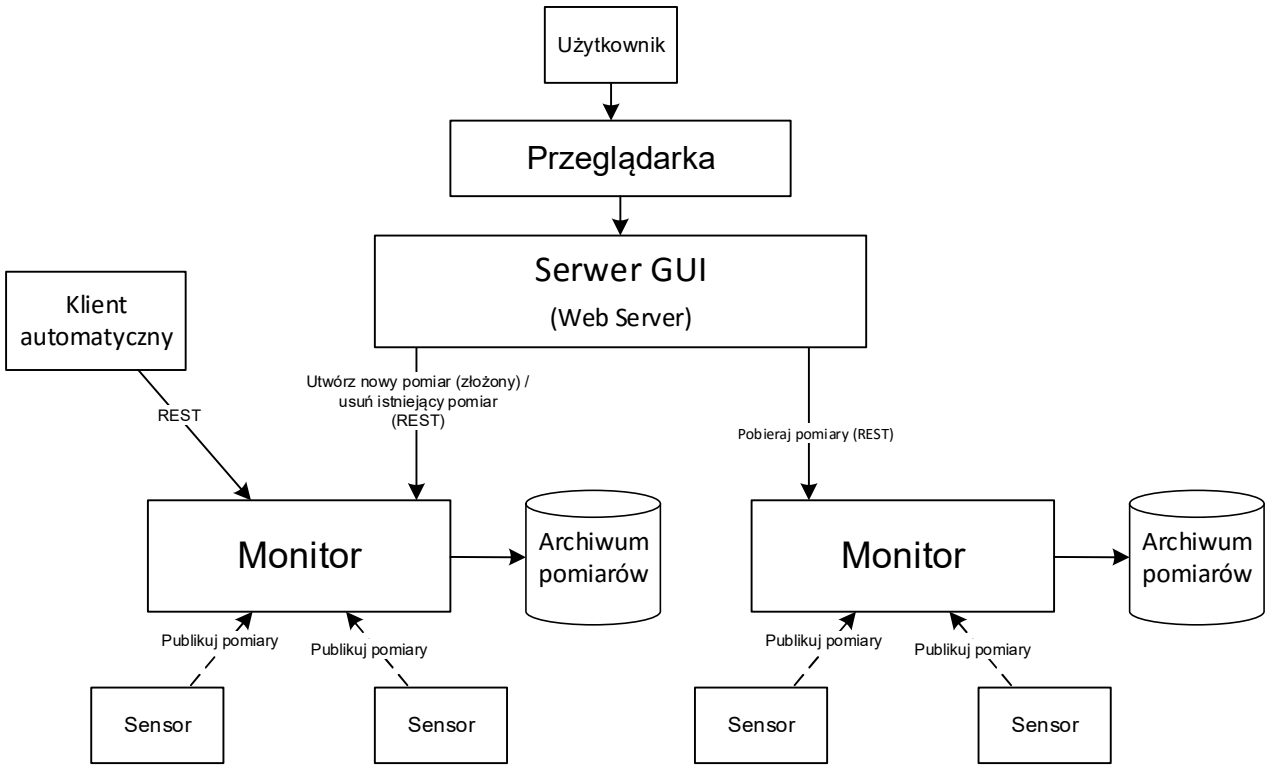
Bartosz Baliś

Programowanie Zespołowe 2016

Temat projektu

System monitorowania rozproszonych zasobów komputerowych, np. obciążenia CPU, zużycia pamięci, obciążenia sieci.

Architektura systemu



Rysunek 1: Architektura systemu do monitorowania zasobów komputerowych.

Główne komponenty systemu, przedstawione na Rys. 1, są następujące:

Sensor

- Dokonuje **pomiarów** wybranych **metryk** dla określonych **zasobów** i wysyła pomiary cyklicznie do Monitora.
  - Przykład zasobu i metryki: *Host* / *CPU Utilization* (aktualne zużycie CPU dla hosta).
- Wiadomość przesyłana przez sensor do monitora powinna zawierać: nazwę (identyfikator) zasobu, nazwę metryki, dane pomiarowe (zależne od metryki).
  - Sensor przesyła dwa rodzaje informacji do monitora: (i) metadane (opisują szczegółowo co jest monitorowane), (ii) właściwe dane pomiarowe. Dane pomiarowe wysyłane są regularnie, natomiast metadane też trzeba przesłać, ale nie ma sensu tego robić za każdym razem gdy przesyłane są dane pomiarowe! (Dla wielu sensorów znacznie zwiększyłoby to obciążenie sieci z powodu monitorowania).
- Wiadomości powinny mieć formę tekstową. Sugerowany format bazowy: JSON.

Monitor

- Zbiera dane od sensorów i **udostępnia pomiary dla klientów**.
- Również umożliwia tworzenie nowych **pomiarów złożonych**, które np. agregują pomiary proste.
  - Pomiary złożone są obliczane w Monitorze na podstawie innych pomiarów.
  - Przykład złożonego pomiaru: *średnie obciążenie CPU z ostatnich 5 minut obliczane co minutę*.

Klient

- **Klient 1**: serwer webowy komunikujący się z Monitorami + przeglądarka sterowana przez użytkownika umożliwiająca wyszukiwanie dostępnych zasobów i pomiarów, wizualne przeglądanie wartości pomiarów i definiowanie nowych pomiarów złożonych.
- **Klient 2**: program automatycznie przeszukujący monitorowane zasoby i pomiary w jakimś celu (np. podobnie do polecenia 'top' wyświetlający top 10 najbardziej obciążonych komputerów).

Archiwum pomiarów

Baza danych, w której przechowywana jest cała historia pomiarów.

Zastosowanie usług REST

Komponenty serwerowe (Monitor) mają udostępniać swoje usługi przez interfejs REST (usługa sieciowa stosująca protokół HTTP). Metodologia tworzenia usług w paradygmacie REST oznacza m.in., że:

1. System rozproszony jest zespołem „zasobów", które mają swoje identyfikatory – URI.
2. Na każdym z zasobów można wykonać tylko cztery proste operacje, tzw. CRUD:
   - **C (Create)**: stworzenie nowego zasobu
   - **R (Read)**: odczytanie stanu zasobu
   - **U (Update)**: zmiana stanu zasobu
   - **D (Delete)**: usunięcie zasobu
3. Powyższe operacje odwzorowują się na metody protokołu HTTP następująco ('<…>' oznacza treść wiadomości ('body') przesyłanej do serwera):
   - `GET {URI}` → pobranie stanu zasobu
   - `POST {URI_kolekcji} <reprezentacja_zasobu>` → utworzenie nowego zasobu
   - `POST {URI_zasobu} <zmiana_stanu>` → (częściowa) zmiana stanu zasobu
   - `PUT {URI} <nowy_stan>` → całkowita podmiana stanu zasobu na nowy
   - `DELETE {URI}` → usunięcie zasobu
4. Wszystkie bardziej złożone operacje są efektem ubocznym zmiany stanu zasobu.
   - Przykładowo operacja `POST http://www.example.com/orders/123 <{"status": "paid"}>` spowoduje zmianę statusu zamówienia na „zapłacono", ale również spowoduje efekty uboczne: przekazanie zamówienia do działu realizacji, nadanie paczki, wysłanie e-maila z powiadomieniem, itp.
5. Serwer nigdy nie powinien przechowywać tzw. stanu sesji (czyli stanu aplikacji). Innymi słowy wszystkie operacje na serwerze są bezstanowe.
   - Przykład: bezstanowy koszyk na zakupy (http://alandean.blogspot.com/2008/11/on-restful-basket-state.html)

Wymagania szczegółowe

1. Klient ma możliwość pobrania z Monitora listy wszystkich monitorowanych zasobów i dostarczanych dla nich pomiarów.
2. Klient może przeszukiwać Monitor przy pomocy zapytań:
   a. wyszukiwać monitorowane zasoby spełniające określone kryteria, np. "wszystkie *hosty*, mają w nazwie ciąg '*zeus*'".
3. Pomiary proste (dostarczane bezpośrednio przez sensory) powinny być udostępniane przez Monitor jako lista wartości. Domyślnie zwracana jest lista kilku ostatnich pomiarów.
4. W przeglądarce można:
   a. Przeglądać listy dostępnych zasobów i pomiarów.
   b. Podglądać ostatnie wartości pomiarów.
   c. Wyszukiwać dostępne zasoby i pomiary (np. po nazwie).
   d. Wyświetlać wybrane pomiary na wykresie, uaktualnianym automatycznie co pewien czas (np. 5 sekund).

5. Klient ma możliwość stworzenia nowego pomiaru (złożonego) w Monitorze.
    a. Ograniczamy się do jednego typu pomiaru złożonego: **agregacji przez obliczanie *średniej ruchomej***
    b. Przykład: *średnie obciążenie CPU z ostatnich 5 minut, obliczane co 1 minutę*. (Długość ruchomego okna czasowego = 5 minut, częstotliwość obliczania = 1 minuta).
    c. Pomiar złożony powinien być dostarczany przez Monitor w ten sam sposób jak prosty (jako lista ostatnich wartości).
6. Pomiar złożony można usuwać.
7. Usunąć pomiar złożony może tylko ten użytkownik, który go stworzył. Oznacza to, że w systemie musi istnieć mechanizm uwierzytelniania i autoryzacji użytkowników.
8. Monitor ma zapisywać historię wszystkich pomiarów do bazy danych.
9. W przeglądarce dodatkowo można:
    a. Podłączać się do wielu Monitorów.
    b. Definiować nowy pomiar złożony jako średnią ruchomą istniejącego pomiaru.
    c. Usuwać wcześniej zdefiniowany pomiar złożony.
    d. Uzyskiwać dostęp do historycznych pomiarów zgromadzonych w Archiwum
        i. Można tworzyć wykresy historycznych pomiarów z zadanego okresu czasu. (Wykresy te są statyczne, tj. nie odświeżają się tak jak wykresy pomiarów bieżących).
        ii. Na jednym wykresie można umieścić wiele pomiarów (oś X jest wspólna, osi Y jest kilka)
        iii. Z gotowego wykresu można importować dane na nim widoczne do formatu CSV.
10. Ma być zaimplementowany klient automatyczny, tj. program, który działa podobnie jak komenda ‘top’: wypisuje i co pewien czas odświeża top 10 najbardziej obciążonych maszyn.
    a. Klient automatyczny może podłączyć się do kilku Monitorów jednocześnie.
    b. Klient automatyczny musi uwzględniać zmiany, np. musi wykrywać dodanie nowych lub usunięcie istniejących maszyn z listy monitorowanych zasobów.

Kamienie milowe i produkty

KM1: 7 kwietnia 2016

Projekt systemu. Jego głównymi elementami ma być projekt interfejsu REST, tj. listy zasobów, ich reprezentacji, a także projekt działania podstawowych scenariuszy użycia systemu z wykorzystaniem interfejsów REST.

Produkty:

- Dokument opisujący projekt API.

KM2: 6 maja 2016

Spełnione są wymagania 1-4. Działa kilka sensorów rozmieszczonych na różnych komputerach, które mierzą co najmniej dwie metryki (np. obciążenie CPU i zużycie pamięci).

Produkty:

- Prototyp systemu + demonstracja działania.

KM3: 9 czerwca 2016

Spełnione są pozostałe wymagania (5-10).

Produkty:

- Prototyp systemu + demonstracja działania.
- Pakiety instalacyjne systemu.
- Podręcznik instalacji.
- Podręcznik użytkownika.