



# Archiwizacja i bezpieczeństwo danych

## Regułowy estymator czasu dostępu systemów HSM

Filip Heryan  
Jakub Jaroszewski

prowadzący: dr inż. Darin Nikolow

## 1 Cel projektu

Celem projektu było stworzenie aplikacji służącej do regułowej estymacji czasu dostępu do pliku umieszczonego w hierarchicznym systemie przechowywania danych. Aplikacja miała działać jako Web Service Globus Toolkit.

## 2 Działanie

Web Service udostępnia jedną operację `estimateRead`. Jej argumentem jest ścieżka do pliku podlegającego estymacji wraz z adresem hosta, na którym jest uruchomiony **HSMMon**, np. `example.com/path/to/file`<sup>1</sup>.

Rezultatem operacji jest obiekt `Estimation` z następującymi polami:

**latency** - opóźnienie odczytu,

**bandwidth** - szerokość pasma,

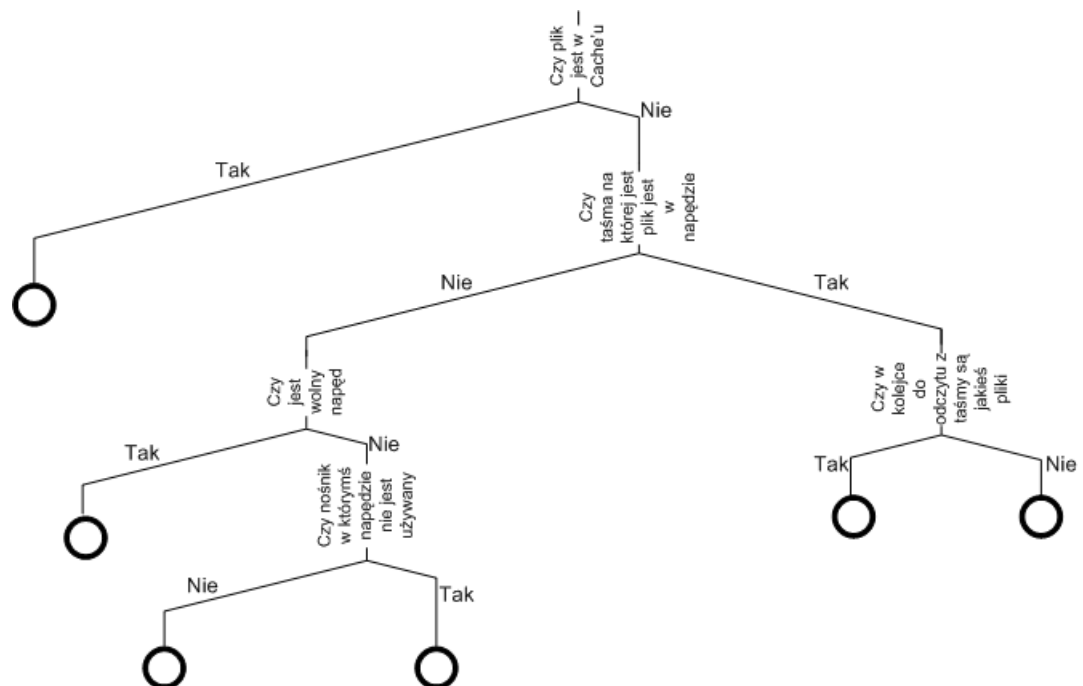
**size** - rozmiar pliku.

---

<sup>1</sup>Zakładamy, że w tym wypadku **HSMMon** jest dostępny pod adresem `http://example.com:8080/wsrf/service/HSMMonFactoryService`.

Estymacja jest oparta na zestawie reguł biorących pod uwagę między innymi:

- zawartość pamięci podręcznej cache,
- taśmy aktualnie dostępne w napędach,
- zajętość napędów,
- kolejkę plików do odczytu.



Rysunek 1: Schemat reguł.

### 3 Konfiguracja

Konfiguracja stałych algorytmu estymacji czasu dostępu do plików odbywa się poprzez plik `properties.xml`<sup>2</sup>. Dostępne są w nim 4 konfigurowalne stałe:

**Positioning latency** - wielkość wyrażona w sekundach – średni czas potrzeby na pozycjonowanie taśmy w napędzie.

<sup>2</sup>Plik powinien znajdować się w `conf/properties.xml` względem katalogu roboczego kontenera Globus Toolkit.

**Load tape latency** – wielkość wyrażona w sekundach – średni czas potrzebny na załadowanie taśmy do pustego napędu.

**Unload tape latency** – wielkość wyrażona w sekundach – średni czas potrzebny na wyciągnięcie taśmy z napędu i zwolnienie ramienia robota.

**Cached latency** – wielkość wyrażona w sekundach – średnie opóźnienie przy pobieraniu pliku już znajdującego się w cache.

Poniżej przedstawiono przykładowy plik konfiguracyjny.

```
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="Positioning latency">60</entry>
<entry key="Load tape latency">10</entry>
<entry key="Unload tape latency">10</entry>
<entry key="Cached latency">1</entry>
</properties>
```

## 4 Instalacja

1. Należy ustawić zmienne środowiskowe konieczne do działania Globus Toolkit: `GLOBUS_LOCATION` oraz `ANT_HOME`.
2. W celu wygenerowania pliku `gar`, należy uruchomić skrypt `build.sh`.
3. Ostatecznie service musi zostać zdeploy'owany w kontenerze Globus Toolkit, można to zrobić przy użyciu skryptu `deploy_ruleest.sh`.
4. Opcjonalnie można zbudować prostego konsolowego klienta korzystającego z service za pomocą skryptu `build_client.sh`.

## 5 Uruchomienie

Po zainstalowaniu wystarczy uruchomić kontener Globus Toolkit. Powinien on wypisać dostępne serwisy. Estymator powinien być dostępny pod adresem `http://example.com:8080/wsrf/services/hsmmon/estimation/rule/EstService`.

Aby ułatwić korzystanie z konsolowego klienta, napisaliśmy skrypt `run_client.sh`. Pierwszym argumentem jest adres serwisu estymatora, kolejnymi ścieżki do plików podlegających estymacji.

Przykład uruchomienia:

```
./run_client.sh \
  http://example.com:8080/wsrf/services/hsmmon/estimation/rule/EstService \
  example.com/path/to/file \
  example.com/path/to/second/file \
  another.example.com/path/to/file
```

Klient wypisze wartości pól zwróconych obiektów `Estimation` dla każdego z plików.