

# Mechanizmy wydajnego zbierania dokumentów internetowych na przykładzie wyszukiwarki NetSprint

# Plan prezentacji

- Główne wyzwania stojące przed wyszukiwarką internetową
- Budowa wyszukiwarki internetowej
- Implementacja wydajnego mechanizmu zbierania dokumentów
- Zbieranie dokumentów internetowych w środowisku rozproszonym

# Wyszukiwarka Internetowa - wyzwania

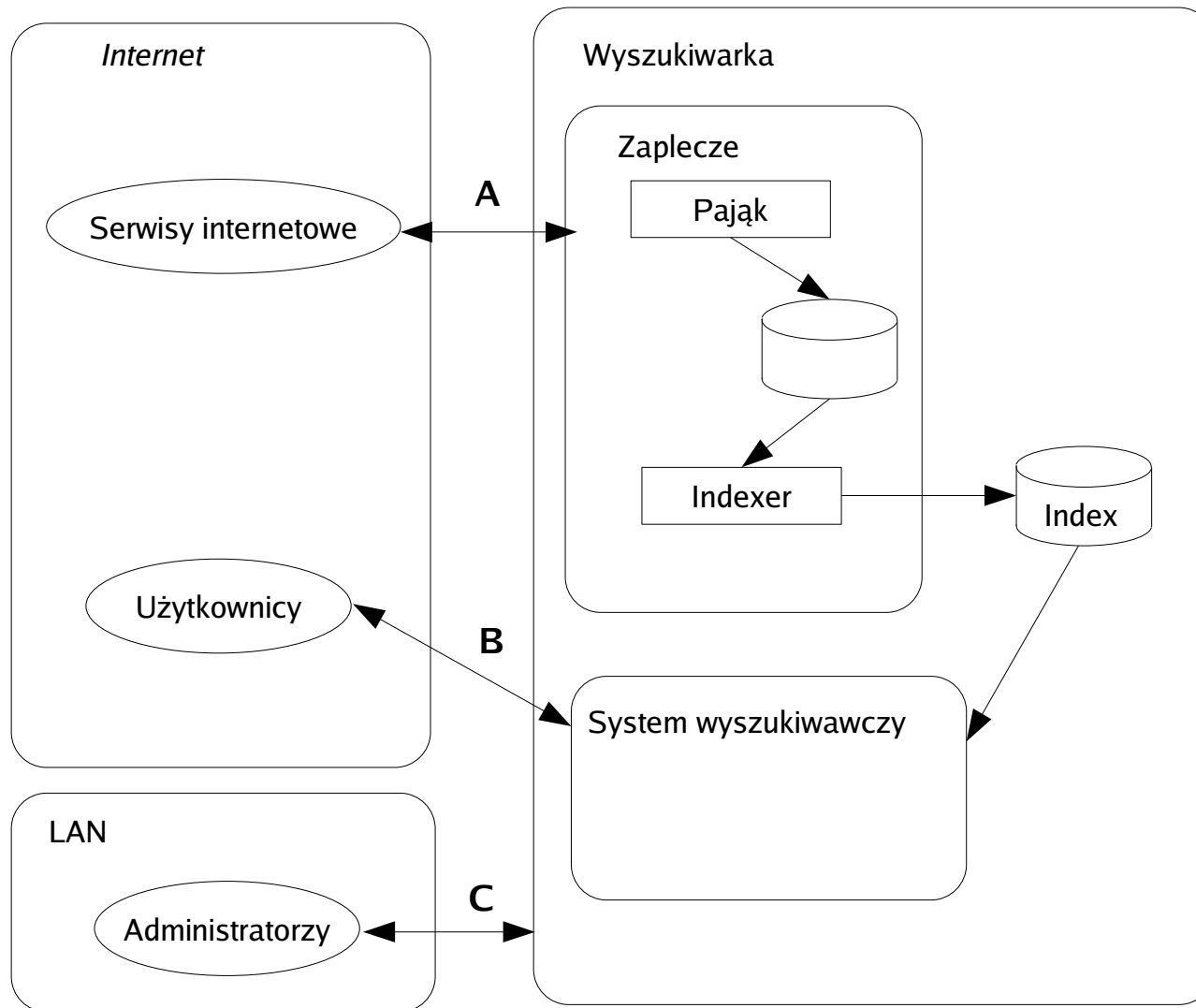
## - Proces gromadzenia danych

- Bardzo duża ilość danych (polski internet zawiera setki milionów dokumentów)
- Nieustannie zmieniające się dokumenty

## - Proces wyszukiwania

- Wydajność rzędu kilkudziesięciu zapytań na sekundę
- Duża odporność na błędy
- Zwracanie wysokiej jakości wyników spośród setek tysięcy dokumentów pasujących do zapytania

# Budowa wyszukiwarki internetowej



## Operacje

- A – pobieranie dokumentów internetowych
- B – zadawanie zapytań do wyszukiwarki
- C – czynności administracyjne

# Komponenty wyszukiwarki - zaplecze

## - Pająk internetowy (*Spider*)

Zadania:

- Wydajne zbieranie dokumentów internetowych (rzędu dziesiątki/s)
- Analiza pobieranych dokumentów oraz składowanie ich w magazynie
- Organizacja kolejki odwiedzanych stron internetowych
- Wykrywanie stron spamerskich

# Komponenty wyszukiwarki - zaplecze

- **Indekser** – moduł odpowiedzialny za generowanie plików indeksu wykorzystywanych przez system wyszukiwawczy.
- Pełna operacja indeksowania wykonywana jest średnio raz na tydzień
- **Pliki indeksu** składają się z:
  - list inwersyjnych dla wszystkich słów wyprasowanych z zebranych dokumentów (dla każdego słowa posortowana lista dokumentów w których to słowo wystąpiło)
  - Pozycje wystąpienia słów w dokumentach – wymagane do wyszukiwania fraz
  - Treść zindeksowanych dokumentów (bez tagów html-owych)

# Komponenty wyszukiwarki - frontend

- **System wyszukiwawczy** – moduł bazujący na wygenerowanych plikach indeksu wyszukujący dokumenty na zadane przez użytkowników zapytania
- Pliki indeksu są zamapowane do pamięci w celu zwiększenia wydajności wyszukiwania (użycie systemu operacyjnego do cachowania najczęściej używanych części indeksu)

# Mechanizm wydajnego zbierania

- Moduł zbierający zaimplementowany w środowisku Java 1.5. Dlaczego Java 1.5:
  - Wykorzystanie wielu standardowych bibliotek (operacje sieciowe)
  - Duża przejrzystość i modułowość kodu
  - Użycie Java generics
  - Wykorzystanie wolnej biblioteki HttpClient z projektu Apache Jakarta

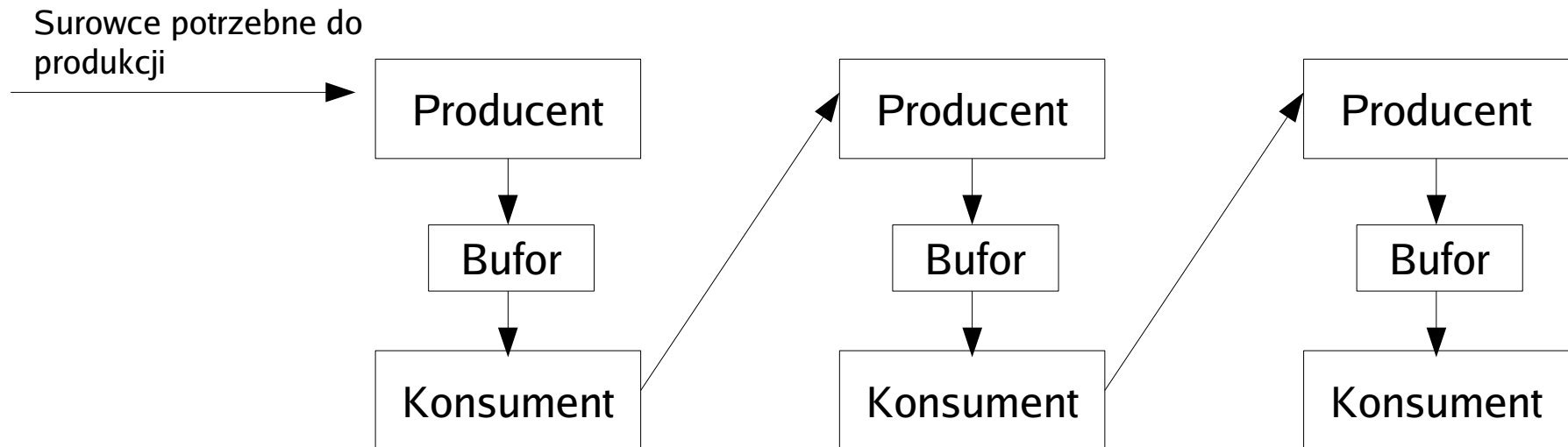
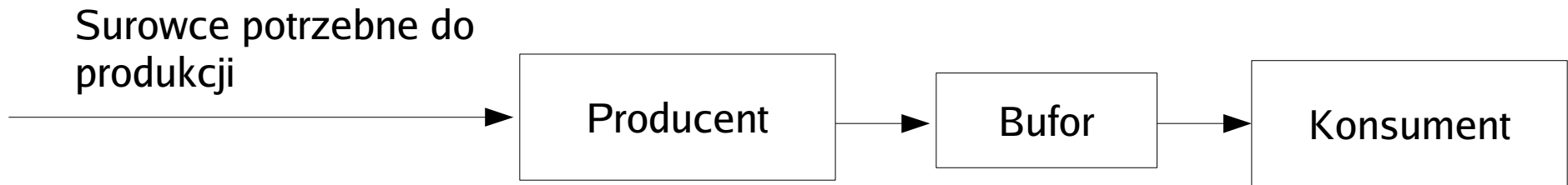
# Mechanizm wydajnego zbierania – problemy

- Niewydajne rozwiązywanie adresów IP za pomocą standardowej biblioteki Java
- Czasochłonne nawiązywanie połączenia z serwerem WWW podczas ściągania pojedynczego url
- Duża ilość danych wymagających utrwalenia (zawartość dokumentu, linki, informacje o nowo znalezionych serwisach)
- Wykonywanie procesu pobierania zawartości dokumentu oraz jego przetwarzania w jednym wątku jest niewydajne (potrzeba tworzenia dużej liczby wątków)
- Zachowanie odstępów czasowych pomiędzy pobraniami stron z tego samego serwisu (około 3 - 5s)

# Architektura modułu zbierającego

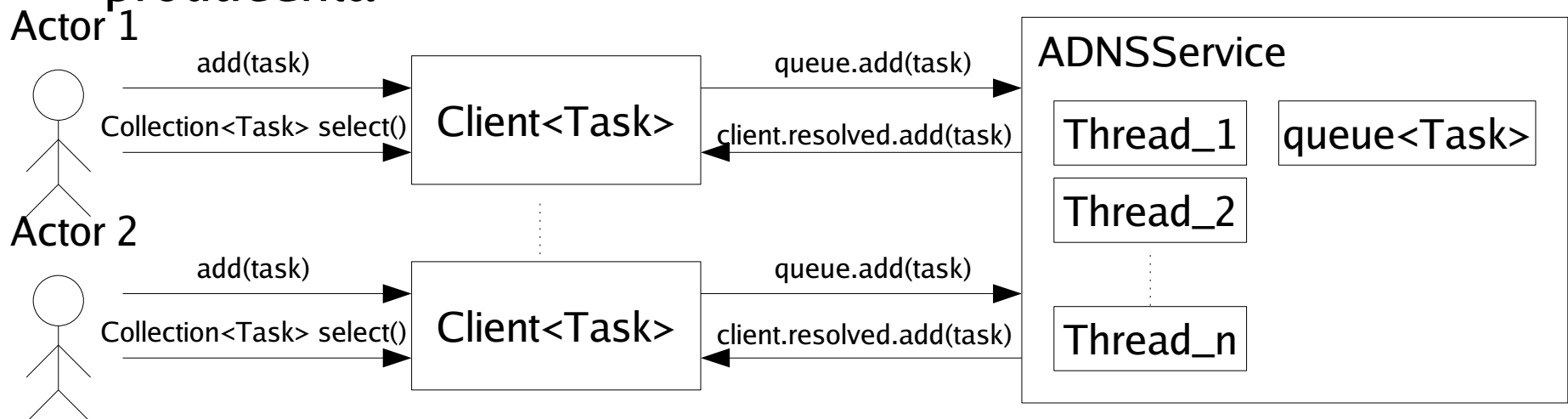
- Architektura systemu zbierającego stworzona została na wzór **linii produkcyjnej**
- Każda część systemu w dużym stopniu funkcjonuje niezależnie
- Dzięki umieszczeniu buforów pomiędzy modułami unikamy przestoju

# Wzorzec projektowy Producent/Konsument



# Komponenty modułu zbierającego

- **ADNSService** – asynchroniczne odpytywanie serwisu DNS.
  - Implementacja wzorca projektowego “producent/konsument”
  - Duża liczba wątków odpytujących DNS-a, każdy rozwiązuje adres pojedynczego serwisu
  - Możliwość podpięcia wielu konsumentów pod jednego producenta

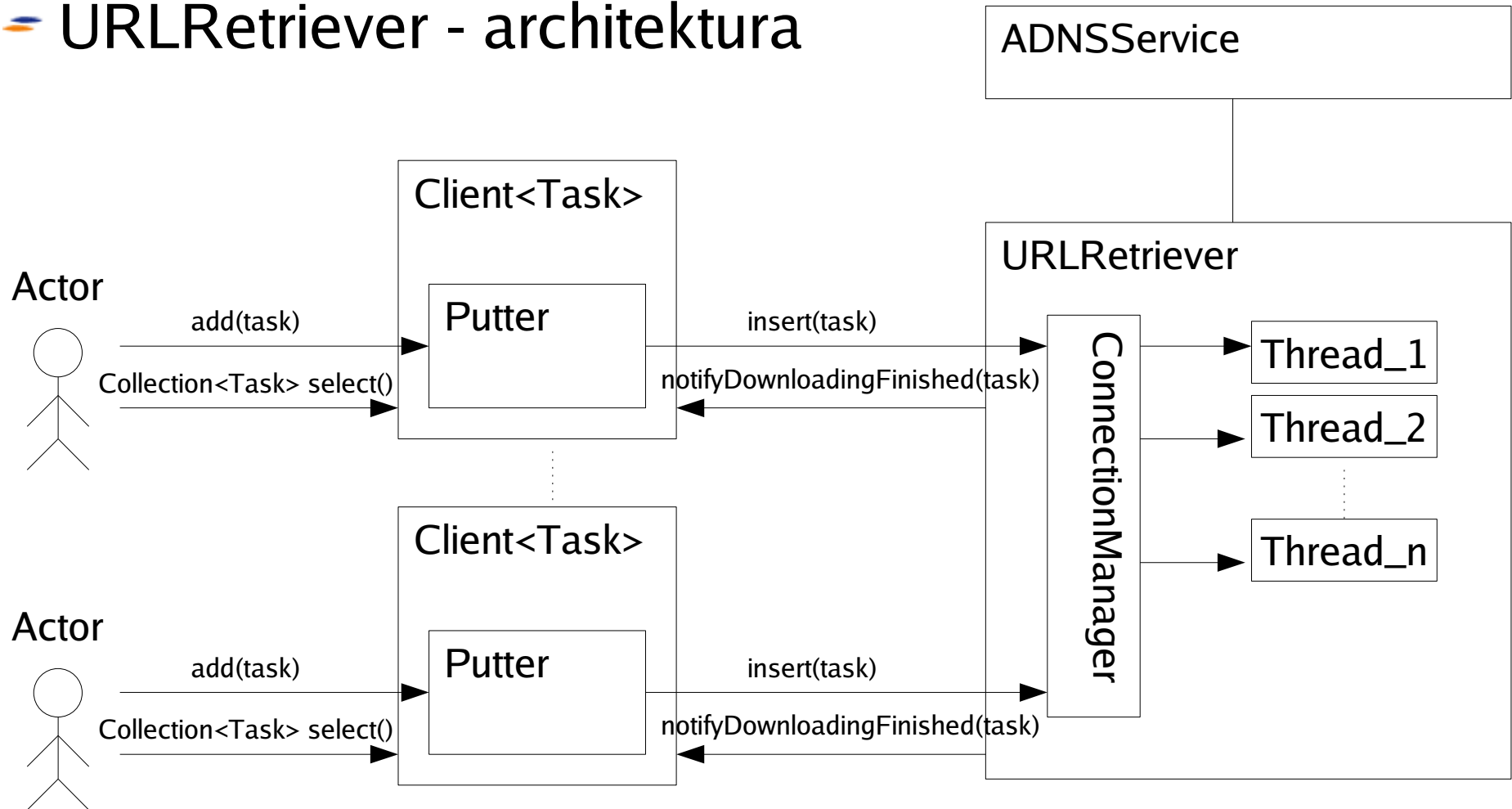


# Komponenty modułu zbierającego

- **URLRetriever** – moduł pobierający dane spod zadanego adresu URL.
  - Implementacja wzorca projektowego “producent/konsument” (możliwość podpięcia wielu konsumentów)
  - Wykorzystuje asynchroniczny DNS
  - Wykorzystuje bibliotekę HttpClient z projektu Apache Jakarta
  - Wielowątkowa architektura
  - Odporność na awarie sieci
  - Odporność na nieprzewidziane zachowanie serwerów WWW

# Komponenty modułu zbierającego

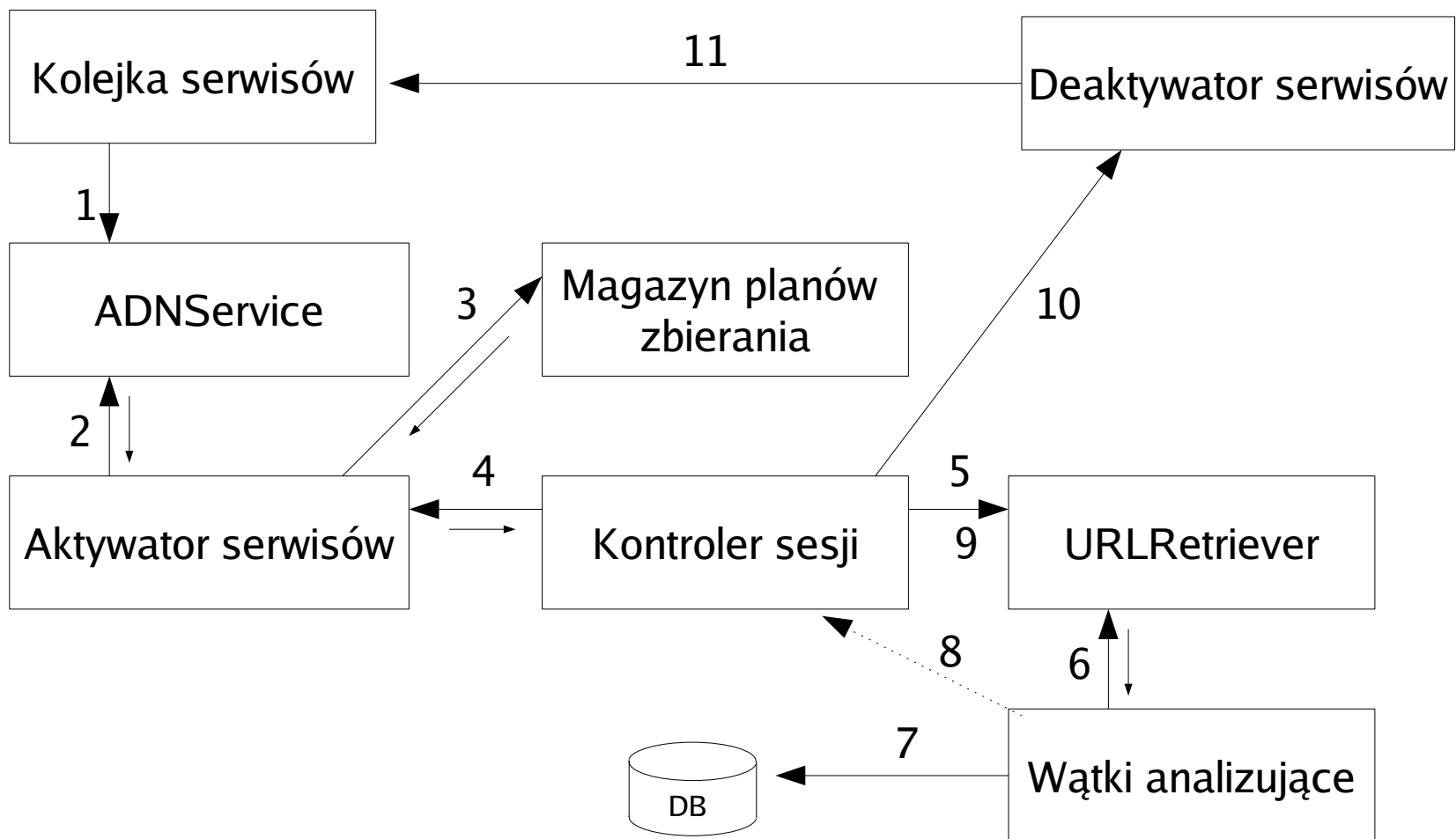
## URLRetriever - architektura



# Komponenty modułu zbierającego

- Zamiast kolejki urli do pobrania przechowywana jest **kolejka serwisów** do odwiedzenia
- Kolejka serwisów jest harmonogramem odwiedzin serwisów
- Każdy serwis posiada swój tymczasowy plan odwiedzin przechowujący kolejki urli do pobrania lub odświeżenia
- **Korzyści:**
  - Jednokrotnie nawiązujemy połączenie z serwerem WWW
  - W pojedynczej sesji pobieramy kilkadziesiąt stron
  - Dużo mniejszy problem grupowania urli z tych samych adresów IP

# Architektura modułu zbierającego

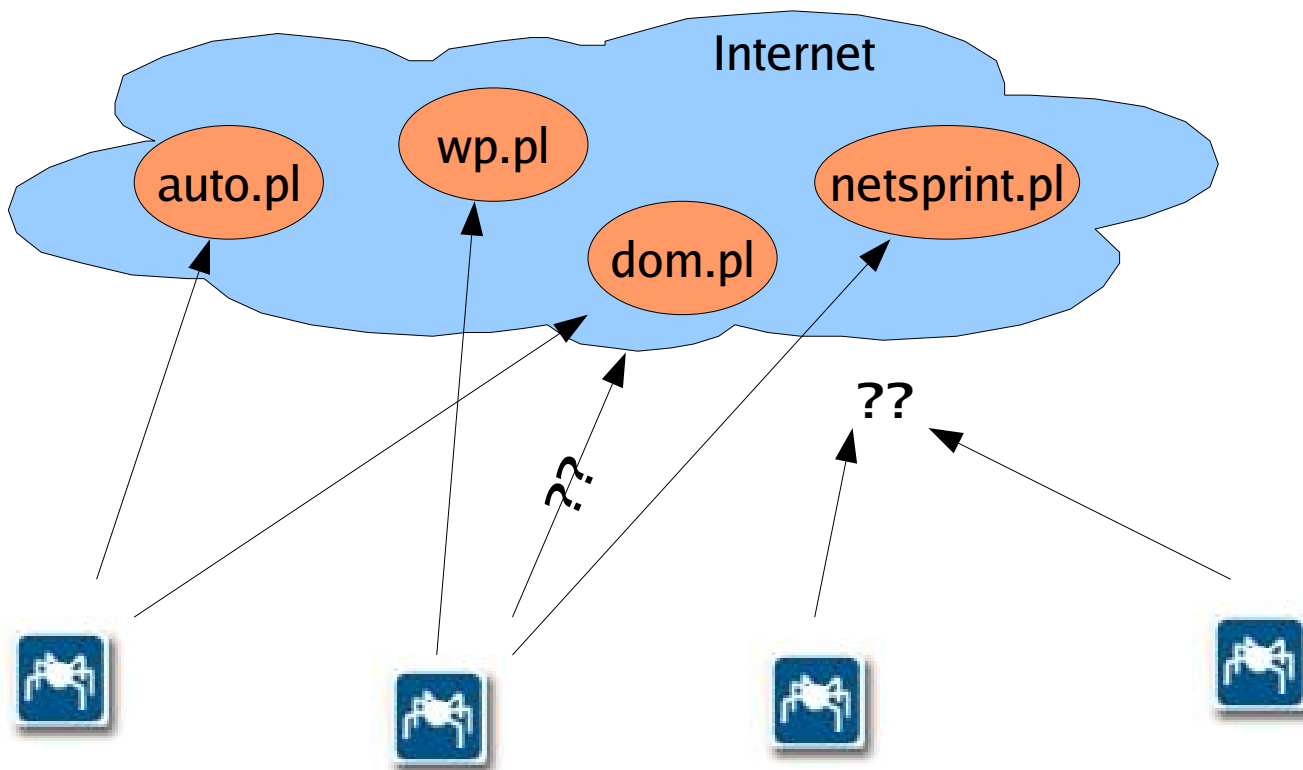


# Mechanizm wydajnego zbierania - podsumowanie

- Prędkość zbierania na poziomie kilkudziesięciu dokumentów na sekundę
- Wymagania pamięciowe na poziomie 1GB
- Duża odporność na nieprzewidziane zachowanie sieci
- Modularność kodu (linia produkcyjna) umożliwia łatwe modyfikacje

# Zbieranie dokumentów internetowych w środowisku rozproszonym - wyzwania

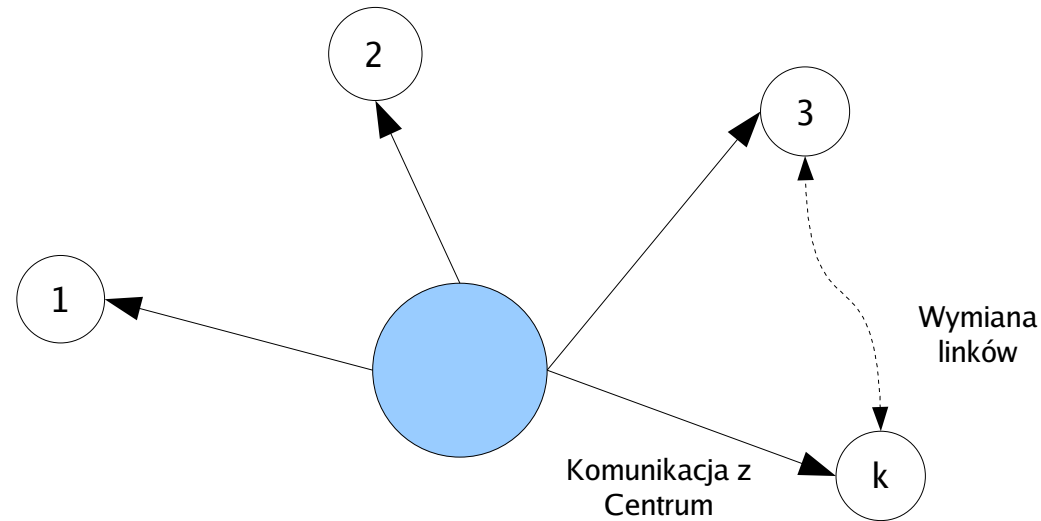
- Jak koordynować prace pajaków rozmieszczonych na niezależnych serwerach



# Zbieranie dokumentów internetowych w środowisku rozproszonym - wyzwania

- Konieczność podejmowania decyzji kto co zbiera
- Wyjściem jest przydzielanie adresów IP poszczególnym serwerom. Adres IP może być obsługiwany tylko przez jednego pająka
- Wynikiem przypisania adresów IP do węzłów jest rozdzielenie pomiędzy nimi serwisów
- Pająki muszą wiedzieć, który serwer obsługuje dany serwis
- Duża ilość danych przesyłanych pomiędzy serwerami (wymiana linków, uzyskiwanie informacji o lokalizacji serwisów)

# Zbieranie dokumentów internetowych w środowisku rozproszonym - architektura



Koordinator przypisujący domeny do węzłów



Węzeł pobierający przypisane do niego dokumenty z sieci

# Zbieranie dokumentów internetowych w środowisku rozproszonym – moduły

- **Centrum koordynujące przydział serwisów** – podejmuje decyzje o przydziale serwisów na podstawie kondycji węzłów (liczba dokumentów, wydajność, itp.).
  - Posiada informacje na temat przydziału adresów IP oraz serwisów do poszczególnych serwerów
  - Centrum może działać samodzielnie, nie wymaga istnienia węzłów

## Centrum koordynujące

Kolejka serwisów  
oczekujących na przydział

Kolejka robotników  
oczekujących na sesje

Przydział adresów IP

Przydział serwisów

# Zbieranie dokumentów internetowych w środowisku rozproszonym – moduły

- **Węzły pobierające dokumenty z sieci (robotnicy)**
  - Posiadają informacje o tym jakie serwisy zostały im przydzielone
  - Posiadają częściową informację o przydzielonych serwisów do poszczególnych węzłów (książka adresowa)
  - Buforują url'e, które powinny trafić na inne węzły
  - Przechowują url'e oraz serwisy, których przydział jest nieznan
  - Mogą pracować przy niedziałającym centrum

## Robotnik

Książka adresowa

Bufor url'i o znanym przydziale

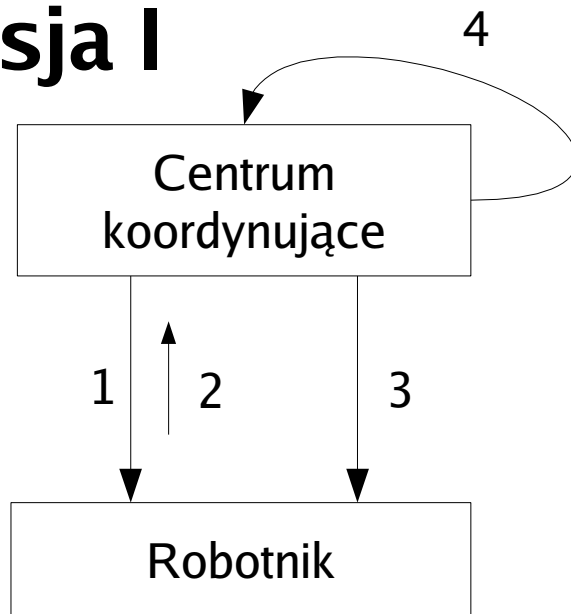
Bufor url'i oraz serwisów bez przydziału

# Zbieranie dokumentów internetowych w środowisku rozproszonym - komunikacja

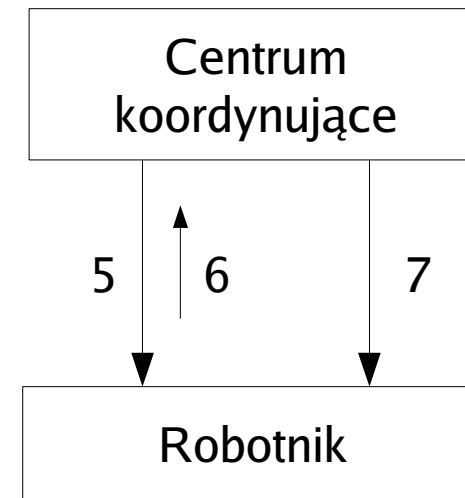
- **Komunikacja pomiędzy robotnikami a centrum** – inicjowana jest przez centrum
  - Podczas sesji robotnika z centrum przekazywane są informacje o serwisach, których przydział jest nieznan
  - Potrzebne są co najmniej dwie sesje aby węzeł dowiedział się gdzie powinny trafić serwisy
- **Komunikacja pomiędzy robotnikami**
  - Po zebraniu się odpowiedniej liczby linków robotnik przesyła je do docelowego węzła
  - Robotnik oczekuje potwierdzenia odbioru url, jeśli wystąpił błąd podczas komunikacji nie usuwa wysłanych url

# Zbieranie w środowisku rozproszonym – komunikacja z centrum

## Sesja I



## Sesja II



- 1 – rozpoczęcie sesji przez centrum
- 2 – przekazanie informacji o serwisach, których przydział jest nieznan
- 3 – zwrócenie informacji o przydziale serwisów, których przydział jest znany centrum
- 4 – dodanie nie przypisanych serwisów do kolejki serwisów oczekujących na przydział
- 5 – rozpoczęcie drugiej sesji przez centrum
- 6 – przekazanie informacji o serwisach, których przydział jest nieznan (pewna część serwisów była wysyłana także we wcześniejszej sesji)
- 7 – zwrócenie informacji o przydziale serwisów

# Zbieranie dokumentów internetowych w środowisku rozproszonym - korzyści

- Duża odporność na błędy
  - Serwery mogą działać niezależnie, awaria jednego z robotników nie wpływa na pracę innych
  - Możliwość odbudowy struktur centrum na podstawie informacji pobieranych od robotników
- Satysfakcjonująca wydajność
  - Największy ruch w sieci odbywa się bezpośrednio pomiędzy robotnikami
  - Dzięki buforowaniu url'i, dane przesyłane są paczkami
  - Mniejsze obciążenie centrum koordynującego dzięki wykorzystaniu książek adresowych na robotnikach

# Dziękuję za uwagę

NetSprint.pl Sp. z o. o.  
ul. Bieżanowska 7  
02-655 Warszawa  
[informacje@netsprint.pl](mailto:informacje@netsprint.pl)  
tel. (022) 844 49 90