Instytut Informatyki
Uniwersytet Wrocławski

Dane w sieciach

(i inne historie)

Marcin Bieńkowski

Jak przechowywać dane w sieciach
(strony WWW, bazy danych, ...)
tak, żeby dowolne ciągi odwołań do (części) tych obiektów
mogły być obsłużone małym kosztem?

Znany problem, wiele wariantów.

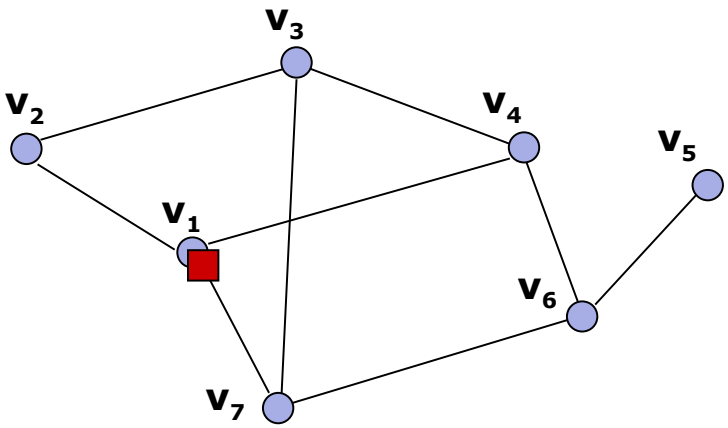
Najbardziej klasyczny i podstawowy z wariantów:

Page Migration

Page Migration (model)

Instytut Informatyki
Uniwersytet Wrocławski

- Sieć n komputerów $v_1, v_2, ..., v_n$



- Jedna kopia jednego niepodzielnego obiektu (strona pamięci) o rozmiarze D w lokalnej pamięci jednego z komputerów
- Koszt komunikacji pomiędzy parą komputerów = = koszt najtańszej ścieżki pomiędzy nimi

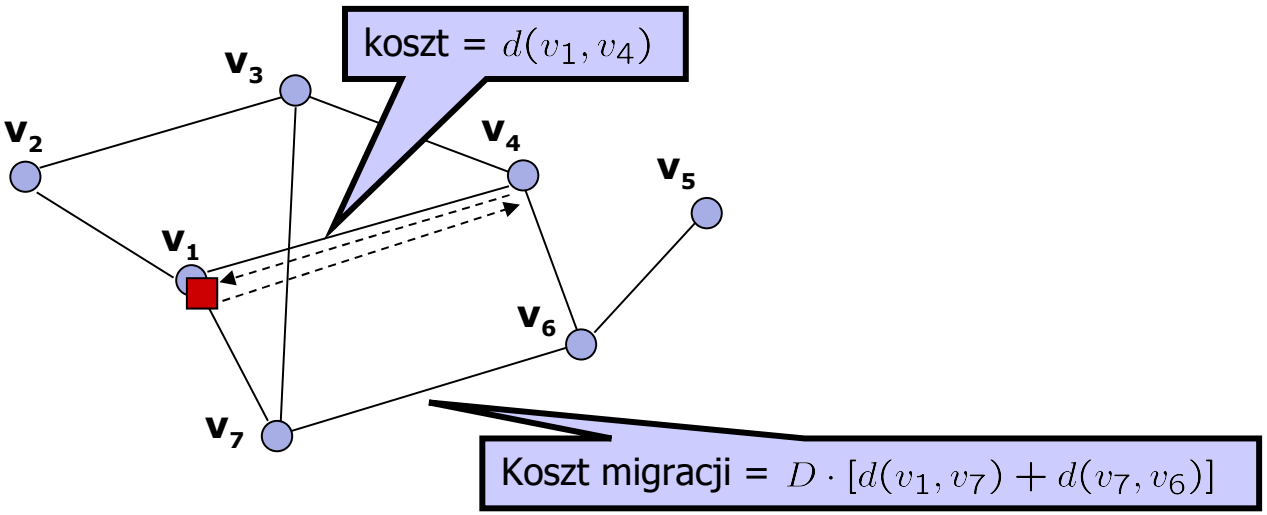
Page Migration (model cd.)

Instytut Informatyki
Uniwersytet Wrocławski

Problem: komputery chcą dostępu do danych ze strony

W jednym kroku t

- σ_t chce odczytać lub zapisać jednostkę danych ze strony



- Po obsłudze żądania algorytm może przenieść stronę do nowego komputera

Problem online

Instytut Informatyki
Uniwersytet Wrocławski

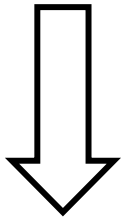
- WEJŚCIE: ciąg $\sigma_1, \sigma_2, \sigma_3, \ldots$
- WYJŚCIE: ciąg migracji strony minimalizujący całkowity koszt

Problem: przyszłość jest nieznana

- Nie mamy szansy wyliczyć optymalnego rozwiązania
- Ale możemy się do niego zbliżyć!
- Istnieją algorytmy (online), które osiągają koszt zaledwie o stałą większy od rozwiązania optymalnego (OPT)

Mój wkład w tematykę

Sieci rzadko są stabilne, zazwyczaj przepustowości łącz zmieniają się w nieprzewidywalny sposób.



Koszty przesyłania danych również ulegają zmianom.

Założenie: Koszty przypisane do krawędzi sieci mogą zmieniać się w dowolny sposób, ale powoli.

Ta wersja problemu okazuje się bardzo trudna

- Każdy algorytm deterministyczny jest co najmniej $\sqrt{D} \cdot n$ razy gorszy od OPT
- Dla algorytmów zrandomizowanych: $\sqrt{D \cdot \log n}$ razy

Umiemy pokazać algorytmy osiągające te współczynniki

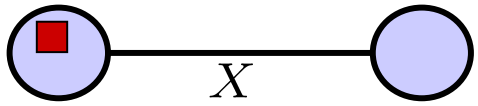
Algorytm dla dwóch wierzchołków

Instytut Informatyki
Uniwersytet Wrocławski

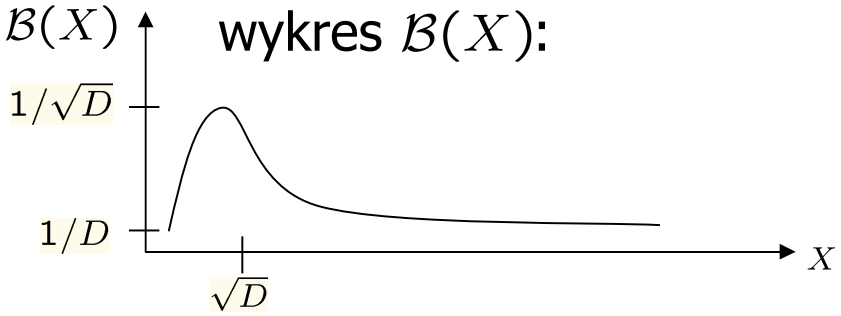
Algorytm EDGE

W każdym kroku przenieś stronę do wierzchołka, który żądał przed chwilą dostępu do niej, z ppb.

$$\mathcal{B}(X) = \begin{cases} \dfrac{X}{D-1+\sum_{i=1}^{X} i} & \text{jesli } X \leq 2D-2 \\ 1/D & \text{w p.p.} \end{cases}$$



wykres $\mathcal{B}(X)$:

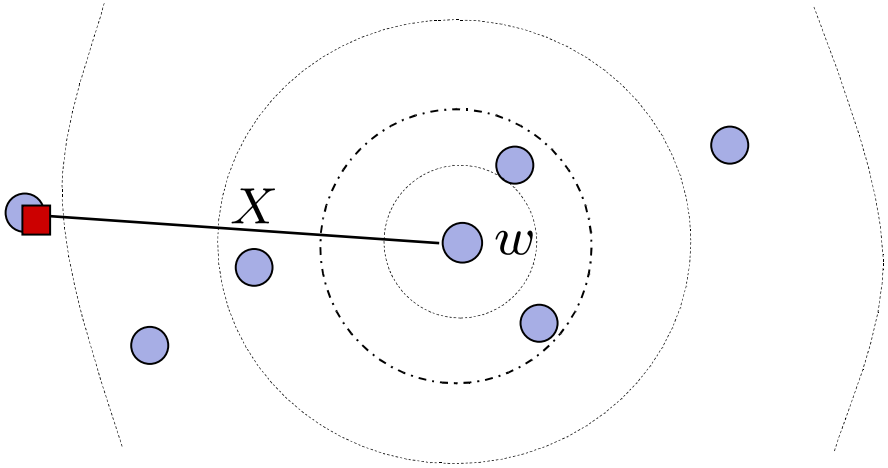


Algorytm dla n wierzchołków

Instytut Informatyki
Uniwersytet Wrocławski

Algorytm DISTRIBUTE

- *Żądanie w wierzchołku* w
- *Przenosimy się z ppb.* $\mathcal{B}(X)$
- *Docelowy wierzchołek wybieramy losowo (im dalej od* w*, tym ppb. mniejsze*



Instytut Informatyki
Uniwersytet Wrocławski

Strategię leżącą u podstaw algorytmu DISTRIBUTE da się lekko zmodyfikować i otrzymać:

Algorytm dla problemu k-SetCover, będący $\mathcal{O}(1 + k/\log n)$-aproksymacją

Inne spojrzenie

Dlaczego przegraliśmy?

Bardzo silny model: adwersarz kontroluje jednocześnie dostęp do strony pamięci jak i zmiany sieci!

Rzeczywistość nie jest tak brutalna
(miejmy nadzieję)

Z drugiej strony:
model całkowicie stochastyczny jest zbyt łatwy

Modyfikacje modelu

Zamieńmy część adwersarza przez
proces stochastyczny

Scenariusz A:

Niech fluktuacje sieci będą losowe, a odwołania do strony najgorsze z możliwych.

Scenariusz B:

Niech zmiany w sieci będą dyktowane przez adwersarza, a komputer czytający ze strony wybierany losowo.

Modyfikacje modelu

Instytut Informatyki
Uniwersytet Wrocławski

Scenariusz A: sieć losowo, żądania adwersarz

$\Rightarrow$ algorytm który jest co najwyżej $\min\{\sqrt[4]{D}, n\} \cdot \text{polylog}(D, n)$ razy gorszy od OPT

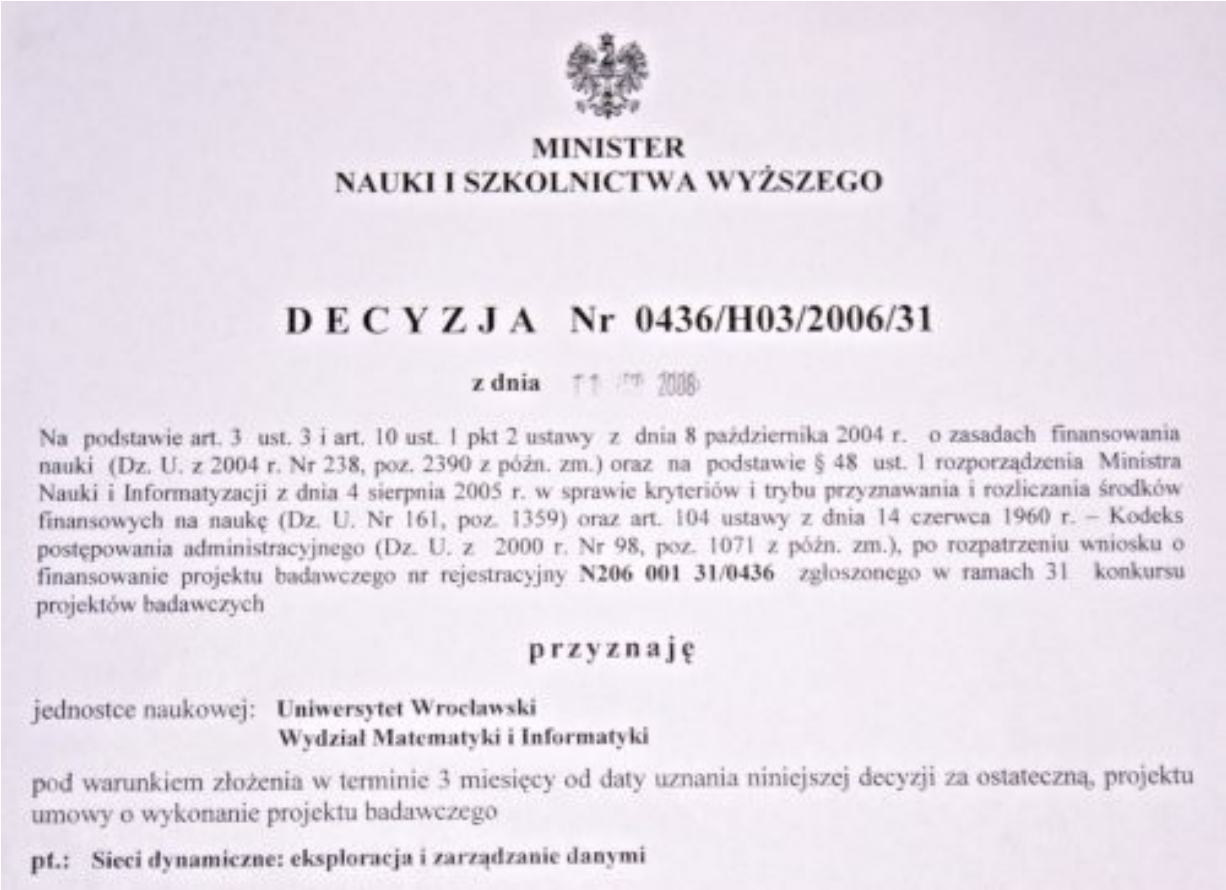
Scenariusz B: sieć adwersarz, żądania losowo

$\Rightarrow$ algorytm który jest co najwyżej o stały czynnik gorszy od OPT

Wyniki osiągane są z dużym prawdopodobieństwem

Co teraz?

- Chciałem zająć się czymś zupełnie innym...
- ... ale minister miał inne zdanie na ten temat



MINISTER
NAUKI I SZKOLNICTWA WYŻSZEGO

D E C Y Z J A Nr 0436/H03/2006/31

z dnia 11 [illegible] 2006

Na podstawie art. 3 ust. 3 i art. 10 ust. 1 pkt 2 ustawy z dnia 8 października 2004 r. o zasadach finansowania nauki (Dz. U. z 2004 r. Nr 238, poz. 2390 z późn. zm.) oraz na podstawie § 48 ust. 1 rozporządzenia Ministra Nauki i Informatyzacji z dnia 4 sierpnia 2005 r. w sprawie kryteriów i trybu przyznawania i rozliczania środków finansowych na naukę (Dz. U. Nr 161, poz. 1359) oraz art. 104 ustawy z dnia 14 czerwca 1960 r. – Kodeks postępowania administracyjnego (Dz. U. z 2000 r. Nr 98, poz. 1071 z późn. zm.), po rozpatrzeniu wniosku o finansowanie projektu badawczego nr rejestracyjny **N206 001 31/0436** zgłoszonego w ramach 31 konkursu projektów badawczych

p r z y z n a j ę

jednostce naukowej: **Uniwersytet Wrocławski**
Wydział Matematyki i Informatyki

pod warunkiem złożenia w terminie 3 miesięcy od daty uznania niniejszej decyzji za ostateczną, projektu umowy o wykonanie projektu badawczego

pt.: Sieci dynamiczne: eksploracja i zarządzanie danymi

Rozszerzenia page migration

- Możliwość replikacji danych (przydatne zwłaszcza przy niemodyfikowalnych obiektach, np. stronach WWW)
- Wiele obiektów w sieci -- komputery mają ograniczoną przestrzeń dyskową

Na szczęście są też inne problemy (choć z podobnej dziedziny) (1)

Instytut Informatyki
Uniwersytet Wrocławski

Eksploracja terenu grupą robotów

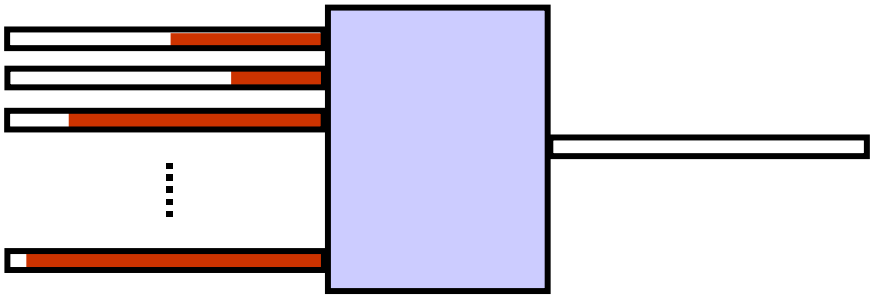
- n robotów
- Komunikacja między robotami
- Każdy wierzchołek grafu musi zostać odwiedzony przez przynajmniej jednego robota

- Chcemy to zrobić jak najszybciej lub też używając jak najmniej energii

Na szczęście są też inne problemy (choć z podobnej dziedziny) (2)

Instytut Informatyki
Uniwersytet Wrocławski

- Bufory w routerach



- m kolejek wejściowych o rozmiarze B
- Jedno wyjście
- Pakiety przychodzące do pełnego bufora są gubione
- Algorytm decyduje, z której kolejki przekazać pakiet na wyjście
- Chcemy zminimalizować liczbę straconych pakietów

Dziękuję za uwagę!