Reorganizacja składu LoXiMa w locie

Marek Dopiera

Plan prezentacji

- Przedstawienie problemu
- Przegląd różnych pomysłów
- Powtórka z MD
- Podejście statystyczne
- Symulowane wyżarzanie
- Stan LoXiMa
- Plan na ten rok

Przedstawienie problemu

- Różnice pomiędzy składem obiektowym a relacyjnym
- Co reorganizować?
- W jakim porządku układać obiekty?
- Kiedy uruchamiać reorganizację?
- Jakie statystyki zbierać?
- Jak wykryć, że aktualna organizacja nie jest optymalna?
- Jak przeprowadzać sam proces reorganizacji?

Przegląd pomysłów

- Praktyczne heurystyki
- Podejście statystyczne
- Wyżarzanie
- Leniwa reorganizacja

Powtórka z MD

- Potrzebne pojęcia:
  - Zmienna losowa
  - Wartość oczekiwana
  - Łańcuch markowa
  - Ergodyczność
  - Rozkład stacjonarny

Podejście stochastyczne

- Definicja modelu
  - Zbiór bloków dyskowych $F = \{0, 1, 2, \ldots, n\}, n \in \mathbb{N}$
  - Zbiór obiektów S
  - Funkcja rozłożenia obiektów po blokach

$$d_\pi \colon S \to F$$

$$d_S \colon F \to F$$

$$d(s) = d_S(d_\pi(s))$$

  - Ciąg zapytań $X_{\{n\}}$
    - 2 warianty
      - Identyczny rozkład, niezależne
      - Łańcuch Markowa
  - Przyjmujemy, że F i S nie zmieniają się

Podejście stochastyczne

- Definicja modelu
  - Czas odczytu zależy wyłącznie od obiektu odczytanego wcześniej $q(x,y)$
  - Czas odczytu n obiektów to

$$T_n = \begin{cases} T_{n-1} + q(X_{n-1}, X_n) & n > 0 \\ 0 & n = 0 \end{cases}$$

  - Średni czas odczytu obiektu to

$$T = \lim_{n\to\infty} \bar{T_n} = E(q(X_{n-1}, X_n))$$

Definicja problemu

- Mając:
  - charakterystykę dostępu do danych
  - Formułę q
- Znaleźć funkcję d, która
  - Minimalizuje T
  - Przypisuje do każdego bloku nie więcej niż L obiektów
  - Być może spełnia inne ograniczenia

Rozwiązania

- Gdy $X_{\{n\}}$ są niezależne o identycznym rozkładzie a $q(X_n, X_{n-1}) = \alpha(1 - \delta_{X_{n-1}, X_n})$ lub $q(X_n, X_{n-1}) = c(|d(X_{n-1}) - d(X_n)|)$ rozwiązania są znane

Rozwiązanie I

- d_π sortuje malejąco obiekty względem ich prawdopodobieństw wystąpienia
- d_S jest identycznością

Rozwiązanie II

- d_π sortuje malejąco obiekty względem ich prawdopodobieństw wystąpienia
- $$d_S(k) = \begin{cases} 2(n-k) & 0 \leqslant k \leqslant n \\ 2(k-n)-1 & n+1 \leqslant k \leqslant 2n \end{cases}$$

Pełniejszy model, łańcuch Markowa

- Na raz działa wielu klientów
- Każdy generuje zapytania o takich samych własnościach probabilistycznych
- Wszystkie zapytania są kolejkowane w jednej kolejce
- Kolejka może być priorytetowa
- Każdy klient ma własny bufor

Pełniejszy model, łańcuch Marokwa

- Szacowanie czasu tak jak w poprzednim wariancie modelu nie przenosi się na ten, więc wprowadzono miarę lokalności dla zapytania WSS, tj dla M odczytów obiektów począwszy od chwili t, przez $K_t^{(M)}$ oznaczamy średnią liczbę różnych odczytów z dysku
- Ze względu a ergodyczność, indeks t, możemy pominąć
- Rozkład zapytań wychodzących z kolejki jest bliski IID

Co chcemy optymalizować?

- Funkcję przypisującą obiekty do bloków, tak, żeby zminimalizować średni rozmiar zbioru roboczego
- Funckję permutującą bloki dyskowe tak, żeby zminimalizować koszt potrzebny serwerowi na odczytanie bloków (już rozwiązane)

Rozwiązanie

- Dla IID proste:

$$K^{(M)} = M - \sum_{q=0}^{|F|-1} (1 - \sum_{y \in d^{-1}(q)} \pi_s(q))^M$$

  - Minimalizowane wtedy, gdy poprzednie problemy
- Dla łańcuchów Markowa
  - Z pytań o obiekty przechodzimy na pytania o bloki dyskowe

Rozwiązanie

- Niech

$$W(M,q) = \begin{cases} 1 & \text{gdy } q \in \{X_1, X_2, \ldots, X_M\} \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

- Wtedy

$$K^{(M)} = \sum_{q=0}^{|F|-1} W(M,q)$$

- Dla M=2 chcemy minimalizować

$$K^{(2)} = 2 - \sum_{q=0}^{|F|-1} \pi_f(q) P_f(q,q)$$

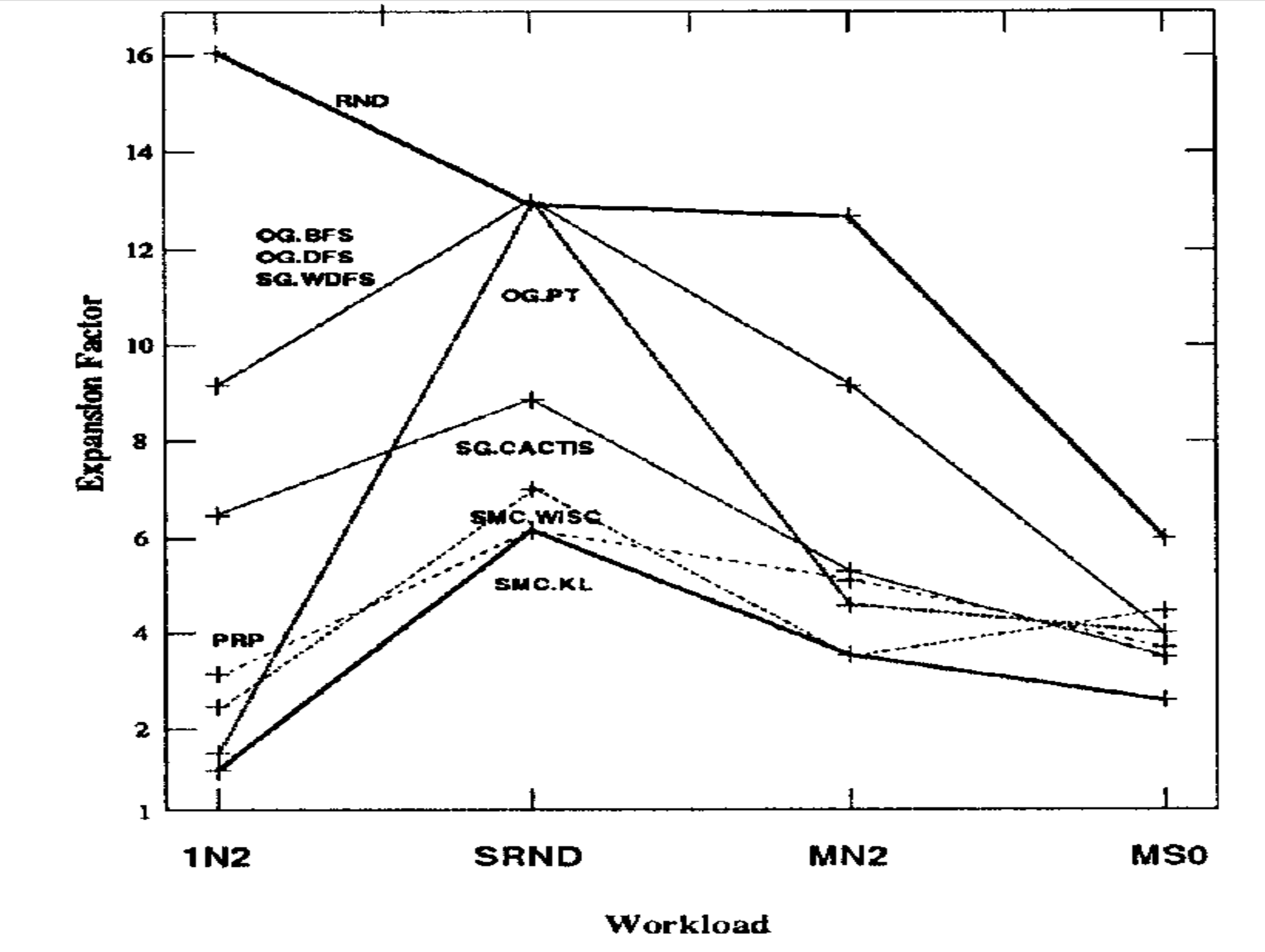
Rozwiązanie

- To jest równoważne minimalizowaniu

$$G_2 = \sum_{q=0}^{|F|-1} \pi_f(q)(1 - P_f(q,q)) = \sum_{q=0}^{|F|-1} \sum_{x \in d^{-1}(q)} \sum_{y : d(y) \neq q} \pi_S(x) P_f(x,y)$$

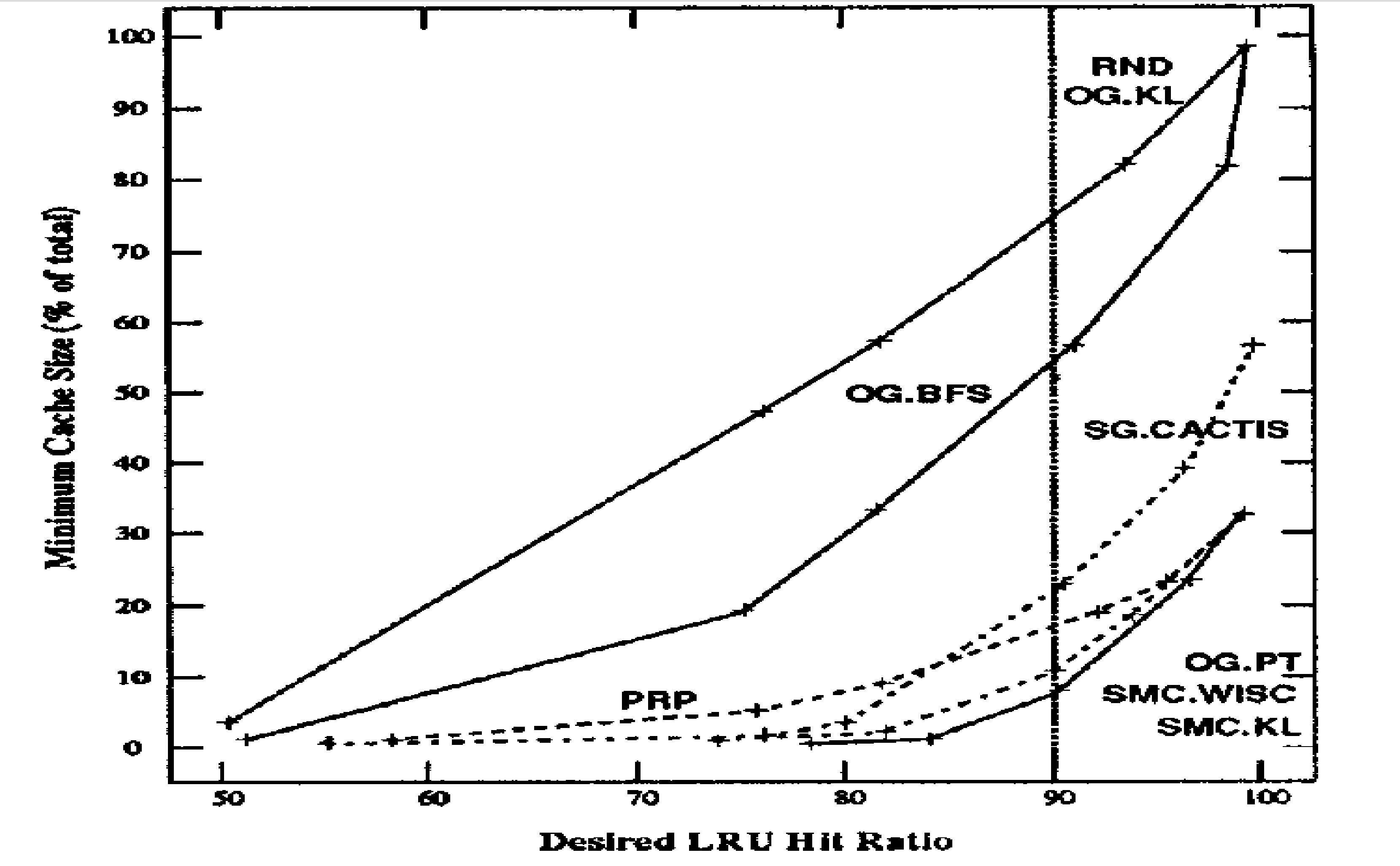
- To jest problem podziału grafu, NP-zupełny ale z dobrymi heurystykami O(n log n)

Testy



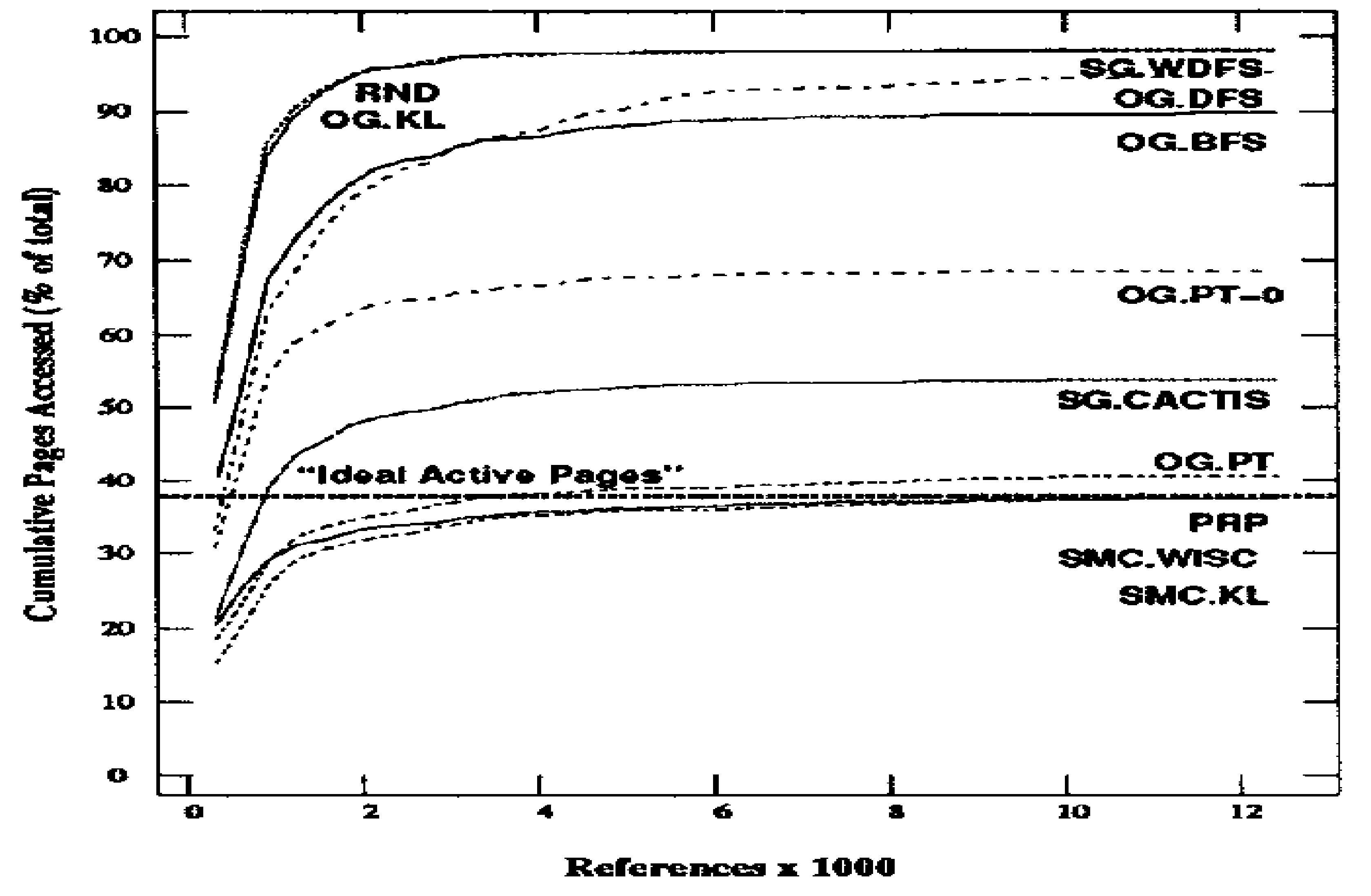
Expansion Factor
16
14
12
10
8
6
4
2
1
RND
OG.BFS
OG.DFS
SG.WDFS
OG.PT
SG.CACTIS
SMC.WISC
SMC.KL
PRP
1N2
SRND
MN2
MS0
Workload

Testy



Minimum Cache Size (% of total)
Desired LRU Hit Ratio
RND
OG.KL
OG.BFS
SG.CACTIS
PRP
OG.PT
SMC.WISC
SMC.KL
0
10
20
30
40
50
60
70
80
90
100
50
60
70
80
90
100

Testy



Cumulative Pages Accessed (% of total)
References x 1000
100
90
80
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12
RND
OG.KL
SG.WDFS
OG.DFS
OG.BFS
OG.PT-0
SG.CACTIS
OG.PT
"Ideal Active Pages"
PRP
SMC.WISC
SMC.KL

Symulowane wyżarzanie

- Kolejne oznaczenia
  - Graf obiektów G(S, A) z funkcją wagową w
  - $$c(d) = \sum_{(o_1,o_2)\in A} w((o_1, o_2))(1 - \delta_{d(o_1),d(o_2)})$$
- Minimalizujemy c poprzez zmodyfikowane symulowane wyżarzanie

Symulowane wyżarzanie

$d \leftarrow$ jakikolwiek naiwny podział grafu G

$T \leftarrow T_0$ **while**(**not** koniec)**do**

    while(**not** równowaga)**do**

        if(zamiana obiektów)

            $d' \leftarrow d$ po losowej zamianie 2 obiektów z różnych bloków

            $\Delta c \leftarrow c(d') - c(d)$

            if($\Delta c < 0$)

                $d \leftarrow d'$

            else

                $d \leftarrow d'$ z prawdopodobieństwem $e^{\frac{-\Delta c}{T}}$

        else

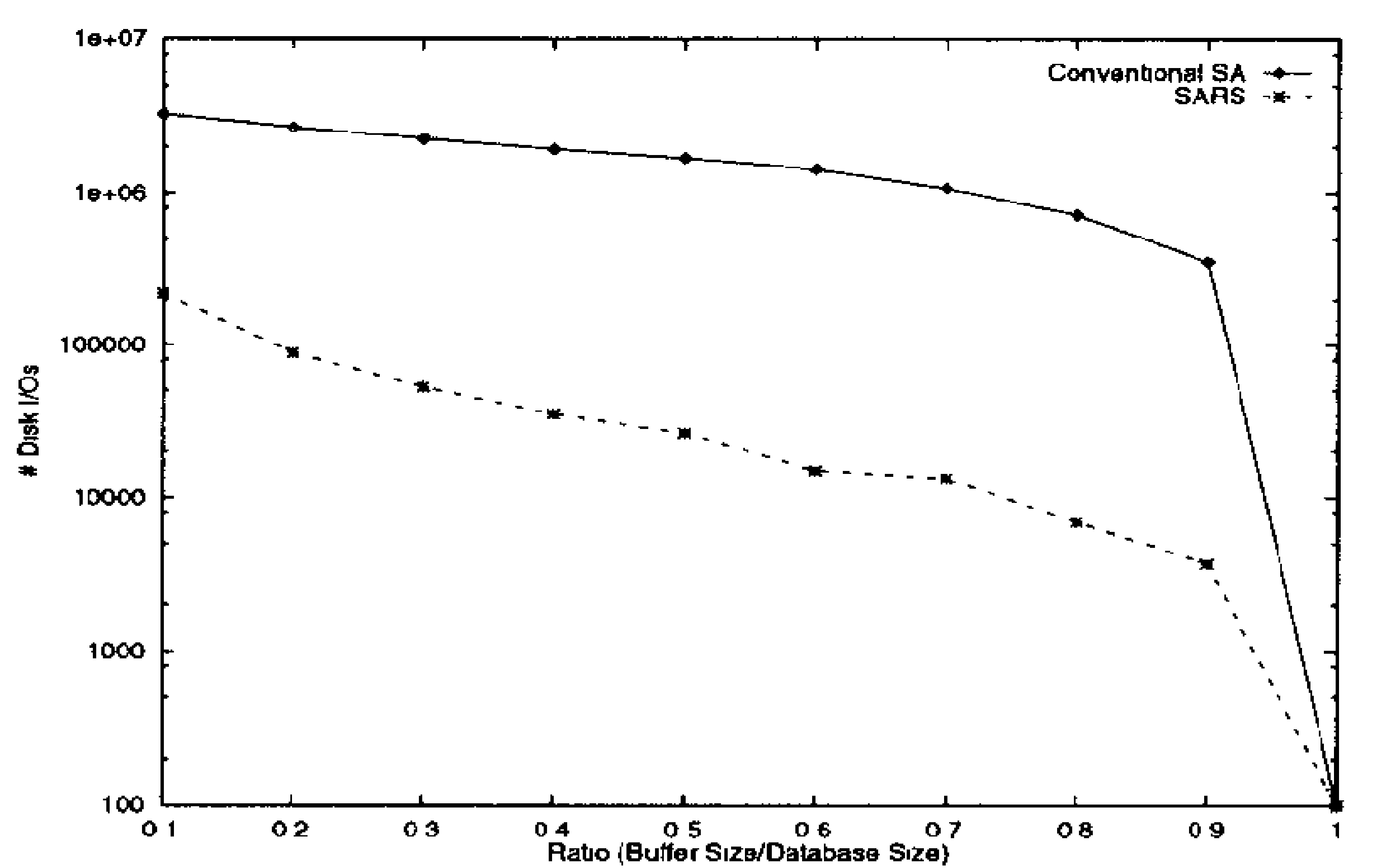
            wczytaj nowy zestaw bloków

    end-while

    $T = \alpha T$

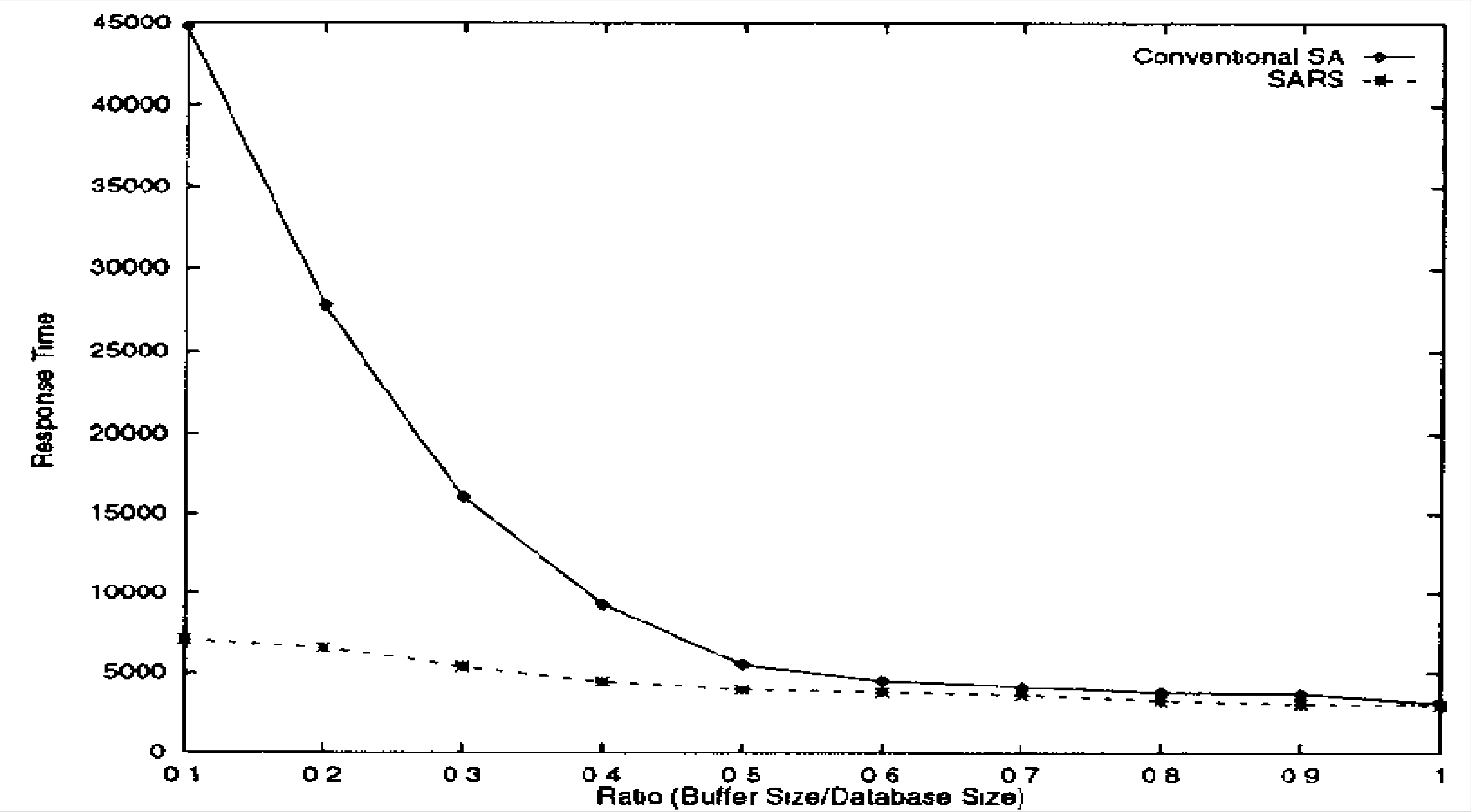
end-while

Testy



1e+07
1e+06
100000
10000
1000
100
Disk I/Os
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
0 9
1
Ratio (Buffer Size/Database Size)
Conventional SA
SARS

Testy



Conventional SA
SARS
Response Time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
0 9
1
Ratio (Buffer Size/Database Size)

Przypuszczalny stan LoXiMa

- Sprawny egzekutor
- Sprawny serwer (poza dużymi danymi)
- Klient do dopracowania
- Skład i logi zmieniane, więc nieistotne
- Wszędzie wycieka pamięć

Plan na ten rok

- Do końca października: porządki organizacyjne z LoXiMem
- Do końca grudnia: postawić LoXiMa na nogi
- Do końca lutego: zaprojektować dokładnie zmiany i wybrać algorytm reorganizacji
- Do końca kwietnia: zaimplementować mechanizm podłączania różnych strategii reorganizacji
- Do końca maja: zaimplementować wybrany algorytm
- Do końca czerwca: testy