On the Practicality of Data-Oblivious Sorting

Kandidatstuderende

Kris Vestergaard Ebbesen

13/4 - 2015

Outline

Outline

Data-Obliviousness

- Hvad er Data-Obliviousness?
- Fordele
  - Branches
  - Hardware
  - Parallisme
- Ulemper
  - Kompleksitet

Data-Oblivious Sorting

- Compare-Exchange
  - Sæt 2 elementer i korrekt rækkefølge
- $\div$ Quicksort, Mergesort, Heapsort ...
- $+$ Bitonic Sort, Pratt's Shellsort ...

Compare-Exchange

procedure COMPARE-EXCHANGE(A, i, j)

$A_{min} \leftarrow \min(A[i], A[j])$

$A_{max} \leftarrow \max(A[i], A[j])$

$A[\min(i, j)] \leftarrow A_{min}$

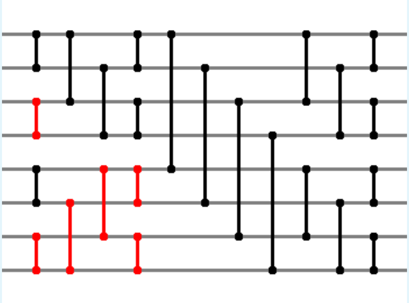
$A[\max(i, j)] \leftarrow A_{max}$

end procedure

Deterministic Data-Oblivious Sorting Algorithms

- Sorteringsnetværk
- Long history
  - Bitonic Sort (68)
  - Odd-Even Mergesort (68)
  - Pratt's Shellsort (72)
  - AKS (83)
  - Zig-Zag Sort (14)
- Problematiske køretider

8-Wire Bitonic Sort



Randomized Data-Oblivious Sorting

- Tilfældigt valgte sammenligninger
- Generelle algoritmer, dybde er ikke så vigtigt
- Nye algoritmer
  - Randomized Shellsort (10)
  - Annealing Sort (14)
- Shaker Sort (87)
- Bedre køretider, ikke garanteret success
- Praktiske problemer

Outline

Outline

Randomized Shellsort

Goodrich, 2010, “Randomized Shellsort: A Simple Oblivious Sorting Algorithm”

- Køretid: $\Theta(n \log n)$
- Fejlrate: $O(n^{-\alpha})$
- Region Comparison
- c og oprydning

Outline

Annealing Sort

Goodrich, 2014, “Spin-the-Bottle Sort and Annealing Sort: Oblivious Sorting via Round-Robin Random Comparisons”

- Køretid: $\Theta(n \log n)$
- Fejlrate: $O(n^{-\alpha})$
- Annealing Sequence
  1. $[(n/2, c), (n/2, c), (n/4, c), (n/4, c) \dots (q \log^6 n, c), (q \log^6 n, c)]$ —- $q \geq 1$ og $c > 1$
  2. $[(q \log^6 n, r), ((q/2) \log^6 n, r), ((q/4) \log^6 n, r) \dots (g \log n, r)]$ — q fra fase 1, $g \geq 1$, og r er $\Theta(\frac{\log n}{\log \log n})$
  3. $[(1, 1), (1, 1) \dots (1, 1)]$ — længde $g \log n$
- Mange konstanter

Outline

Bitonic Sort

Batcher, 1968, "Sorting Networks and Their Applications"

- Klassisk Sorteringsnetværk
- Bitoniske Sekvenser
- Køretid: $\Theta(n \log^2 n)$
- Dybde: $\Theta(\log^2 n)$

Outline

Odd-Even Mergesort

Batcher, 1968, "Sorting Networks and Their Applications"

- Klassisk Sorteringsnetværk
- Merging A and B
  - $C = a_0, (b_0, b_1), (b_2, a_1), (a_2, b_3) \ldots$
  - $C = b_0, (b_1, a_0), (b_2, a_1), (b_3, b_4) \ldots$
- Køretid: $\Theta(n \log^2 n)$
- Dybde: $\Theta(\log^2 n)$

Outline

Shell, 1959, "A High-speed Sorting Procedure"

- Shellsort
  - Mulig, men ikke egnet
  - Mere et framework
- Pratt's Shellsort
  - $2^i 3^j < n$ sekvens
  - Ét enkelt løb per indgang i sekvensen, $\Theta(n \log^2 n)$
- Shaker Sort
  - $\lfloor 1.7^j \rfloor + 1 < n$ sekvens
  - Enkelt løb, op og ned, $\Theta(n \log n)$
  - Ukendt fejlrate

Outline

Zig-Zag Sort

Goodrich, 2014, "Zig-zag Sort: A Simple Deterministic Data-oblivious Sorting Algorithm Running in $O(N \log N)$ Time"

- Deterministisk $O(n \log n)$ netværk, $\Theta(n \log n)$ *dybde*
- Præcist beskrevet
- Meget bedre konstanter end AKS, men meget dybt
- Afhængig af ϵ-halvers, men bedre ϵ

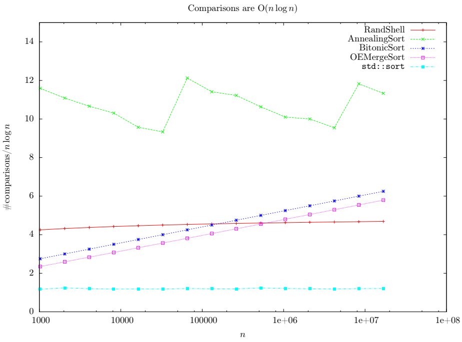
Zig-Zag Sort

- Faktiske tal: $\approx 50cn \log n$, hvis ϵ-halver er cn
- Praktisk implementation, $n = 1024$, #sammenligninger:
  - Zig-Zag Sort - 7056780
  - Odd-Even Mergesort - 24063

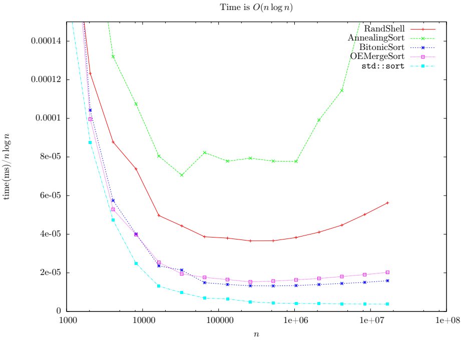
Outline

Nye Algoritmer

Sammenligninger

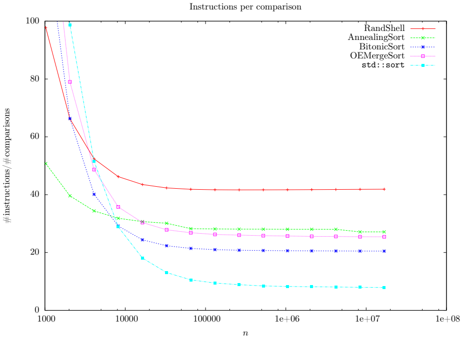


Tid

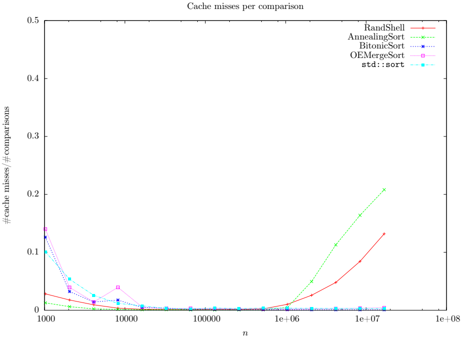


Nye Algoritmer

Instruktioner

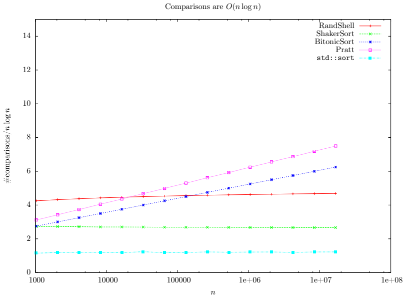


Cache-Misses

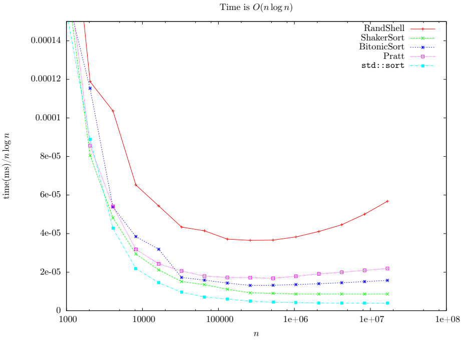


Shellsorts

Sammenligninger

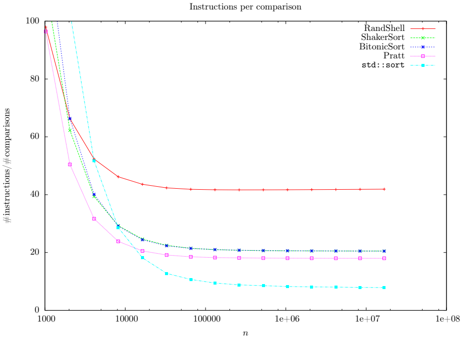


Tid

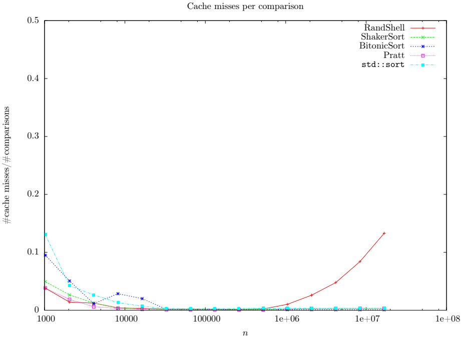


Shellsorts

Instruktioner



Cache-Misses

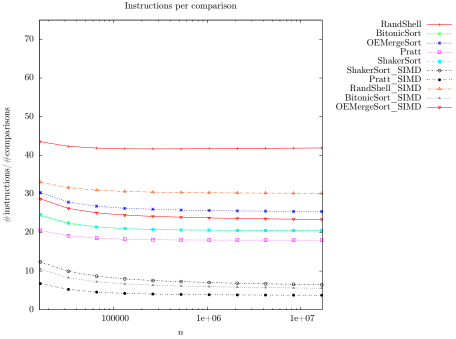


SIMD

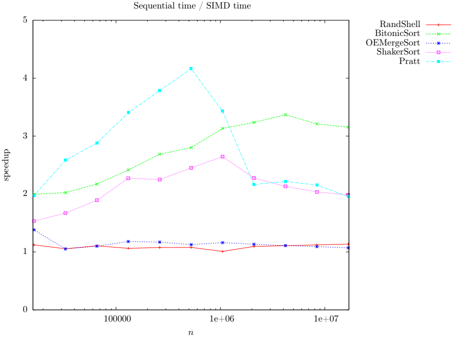
- SSE4.1 - 128 bit, 4x32 bit
  - Registre
  - `PMINSD / PMAXSD`
- Data Alignment
  - 16-byte aligned
  - 16-byte unaligned
  - Individuelle loads
- Brugbart? Ja

SIMD

Instruktioner



Tidsændring



CUDA

- **C**ompute **U**nified **D**evice **A**rchitecture
- Data-Obliviousness

Individuel Tilpasning

Randomized Shellsort CPU -> Texture Shuffle

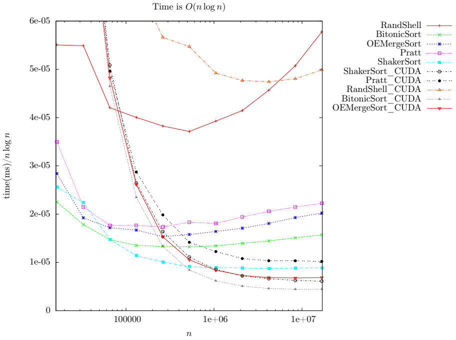
Bitonic Sort Wire Mapping, Shared memory

Odd-Even Mergesort Speciel Remapping

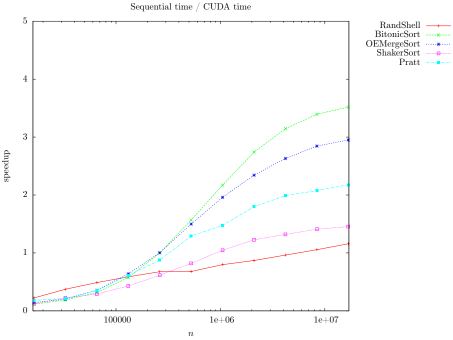
Shellsort Varianter 1 tråd per sub-sekvens

CUDA - Quadro FX 880M

Tid

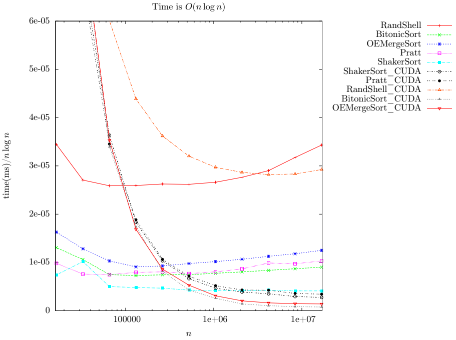


Tidsændring

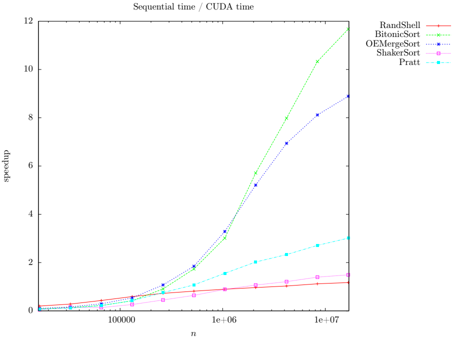


CUDA - GTX 880M

Tid



Tidsændring



OpenMP

- OpenMP Basics
- `#pragma omp ...`
- Stort overhead

Individuel Tilpasning

Randomized Shellsort 1 tråd shuffler, mange sammenligner

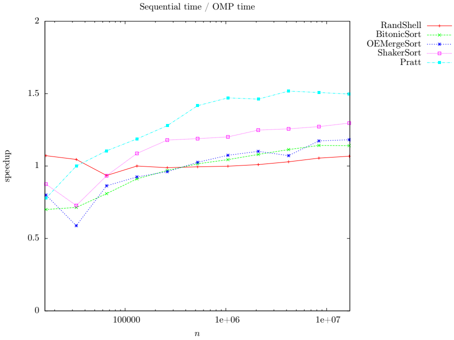
Bitonic Sort Tasks

Odd-Even Mergesort Tasks

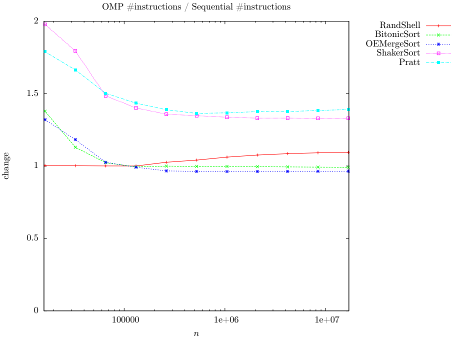
Shellsort Varianter Manuel scheduling grundet cache

OpenMP

Tidsændring

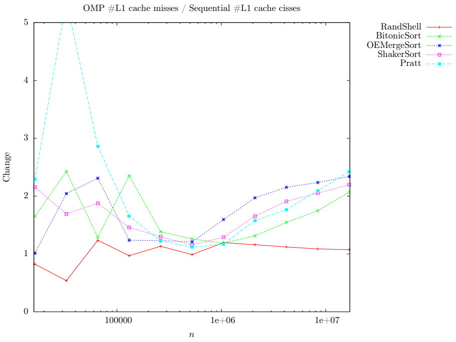


Instruktionsændring

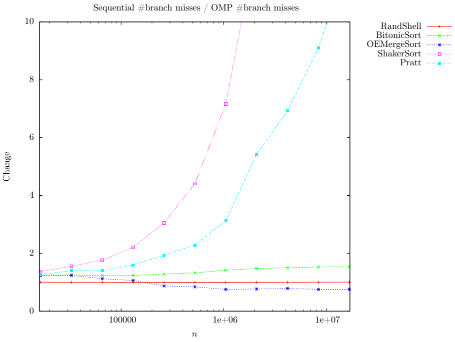


OpenMP

Cacheændring



Branchændring



Outline

- De nye algoritmer er smarte, men fungerer dårligt i praksis
- Ikke meget at gøre ved det
- Men! Nye teknikker gør de gamle algorithmer hurtige

	Base	SIMD	CUDA	OpenMP
Randomized Shellsort	22.6	20.2	**20.1**	21.5
Bitonic Sort	6.41	2.02	**1.80**	5.61
Odd-Even Mergesort	8.15	7.60	**2.76**	7.02
Pratt's Shellsort	8.82	4.50	**4.11**	5.85
Shaker Sort	3.48	**1.75**	2.46	2.65
Annealing Sort	**67.3**	-	-	-