

Systemnahe Informatik

Übungsgruppe Xeon Phi

Dominik Walter

Sommersemester 2018

Zugriffsmuster auf eine Matrix

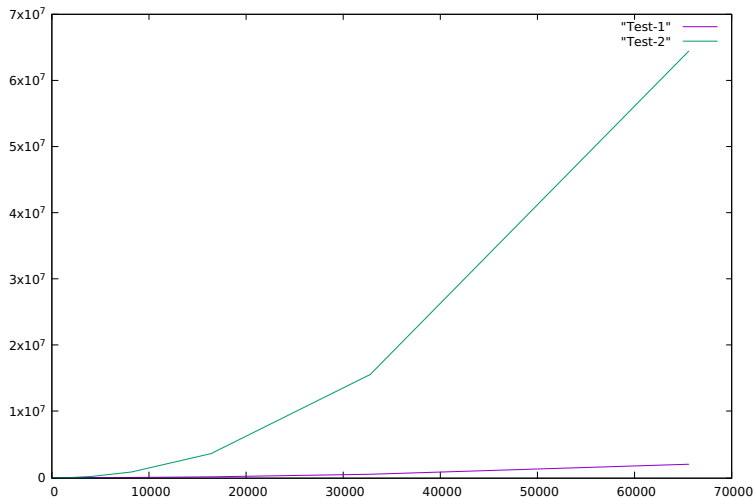
Zeilenweises Iterieren

```
for (int y = 0; y < size; y++) {  
    for (int x = 0; x < size; x++) {  
        matrix[y * size + x] = x * y;  
    }  
}
```

Spaltenweises Iterieren

```
for (int x = 0; x < size; x++) {  
    for (int y = 0; y < size; y++) {  
        matrix[y * size + x] = x * y;  
    }  
}
```

Zugriffsmuster auf eine Matrix



Zugriffsmuster auf eine Matrix

8 Cache-Lines mit jeweils 8 Byte

(0, 0) 0x00	(1, 0) 0x04	(2, 0) 0x08	(3, 0) 0x0C
Tag: 0x0,0		Tag: 0x0,1	
(0, 1) 0x10	(1, 1) 0x14	(2, 1) 0x18	(3, 1) 0x1C
Tag: 0x1,0		Tag: 0x1,1	
(0, 2) 0x20	(1, 2) 0x24	(2, 2) 0x28	(3, 2) 0x2C
Tag: 0x2,0		Tag: 0x2,1	
(0, 3) 0x30	(1, 3) 0x34	(2, 3) 0x38	(3, 3) 0x3C
Tag: 0x3,0		Tag: 0x3,1	

Zeilenweise

0x0,0 → 0x0,0 → 0x0,1 → 0x0,1
0x1,0 → 0x1,0 → 0x1,1 → 0x1,1
0x2,0 → 0x2,0 → 0x2,1 → 0x2,1
0x3,0 → 0x3,0 → 0x3,1 → 0x3,1

Spaltenweise

0x0,0 → 0x1,0 → 0x2,0 → 0x3,0
0x0,0 → 0x1,0 → 0x2,0 → 0x3,0
0x0,1 → 0x1,1 → 0x2,1 → 0x3,1
0x0,1 → 0x1,1 → 0x2,1 → 0x3,1

std::vector vs. std::list

std::vector

```
std::vector<int> vector;  
for (auto x: vector) {  
    result += x;  
}
```

std::list

```
std::list<int> list;  
for (auto x: list) {  
    result += x;  
}
```

Array vs. Liste

Array

```
start:
beq      t0, a2, end

lw       t3, 0(a1)
add      a0, a0, t3

add      a1, a1, 4
add      t0, t0, 1
j start
end:
```

Liste

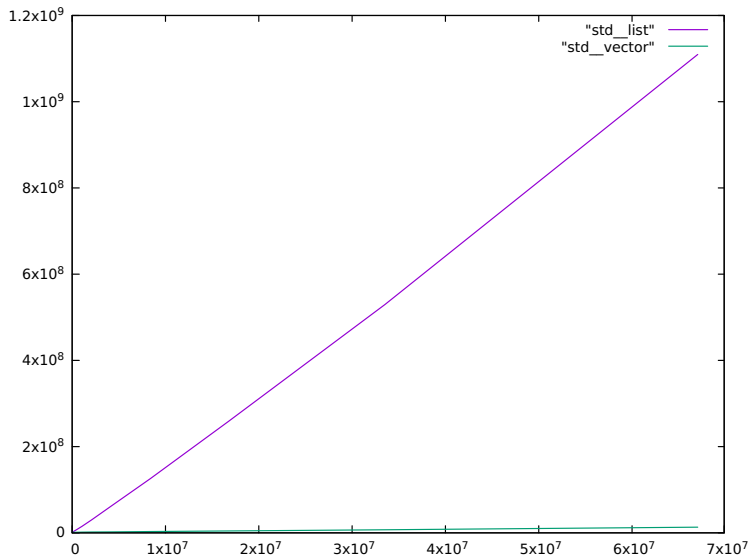
```
# struct node {
#         node* next;
#         int data;
# }

start:
beq      a1, zero, end

lw       t0, 4(a1)
add      a0, a0, t0

lw       a1, 0(a1)
j start
end:
```

std::vector vs. std::list



Zugriffsmuster eines Arrays

2 Cache-Lines mit jeweils 8 Byte

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	a	c	h	e	s		s	i	n	d		t	o	l	l
Tag: 0x0,0								Tag: 0x0,1							

Zugriffsmuster:

0x0,0 → 0x0,0 → 0x0,0 → 0x0,0 → 0x0,0 → 0x0,0 → 0x0,0 → 0x0,0
→ 0x0,1 → 0x0,1 → 0x0,1 → 0x0,1 → 0x0,1 → 0x0,1 → 0x0,1 → 0x0,1

Zugriffsmuster einer Liste

Hinweis: Hier nur 8 Bit pro Pointer!

18	C	q	s	22	n	x	d	26	s	e	f	r	b	c	y
Tag: 0x0,0								Tag: 0x0,1							
i	k	36	a	g	q	40	d	v	m	44	-	t	o	h	w
Tag: 0x1,0								Tag: 0x1,1							
50	l	c	h	54	c		s	58	-	d	k	62	s	o	l
Tag: 0x2,0								Tag: 0x2,1							
f	a	0	l	e	s	72	h	i	n	76	t	h	q	4	i
Tag: 0x3,0								Tag: 0x3,1							
b	g	z	u	j	k	d	w	8	e	c	t	32	o	a	p
Tag: 0x4,0								Tag: 0x4,1							

Zugriffsmuster: (jeweils 2x)

0x0,0 → 0x1,0 → 0x2,0 → 0x3,0 → 0x4,1 → 0x0,1 → 0x1,1 → 0x2,1
→ 0x3,1 → 0x0,0 → 0x1,0 → 0x2,1 → 0x3,1 → 0x4,1 → 0x2,0 → 0x3,0