Alexandru Calotoiu

DaFlEx

C2DaCe challenges

- **Classes**
  - Inheritance
  - Contexts
- **Recursions**
  - Tail recursion
  - Indirect recursion
- **Pointers**
  - Unrestrictred arithmetic
- **Stateful library calls**
  - Automatic assessment
- **Template programming**
- **Library nodes**
- **Encapsulation**

F2DaCe challenges

- **Generalized views**
- **Vector operations**
- **Labels & GoTo's**
- **Intrinsic function coverage**
- **Modern Fortran**

DaCe challenges

- **Application-level ToGPU transform**
  - + Associated transforms

Engineering efforts

Research efforts

Application-level ToGPU transform

Map Fission

- On any SDFG
- Must handle
  - Edge assignments
  - Scalars
  - Control flow

Map Fusion

- On any pair of Maps
- Must accept
  - Conditions
- Needs helper Transformations

Map to GPU

- Must work on arbitrary Maps
- Not a state-level transformation

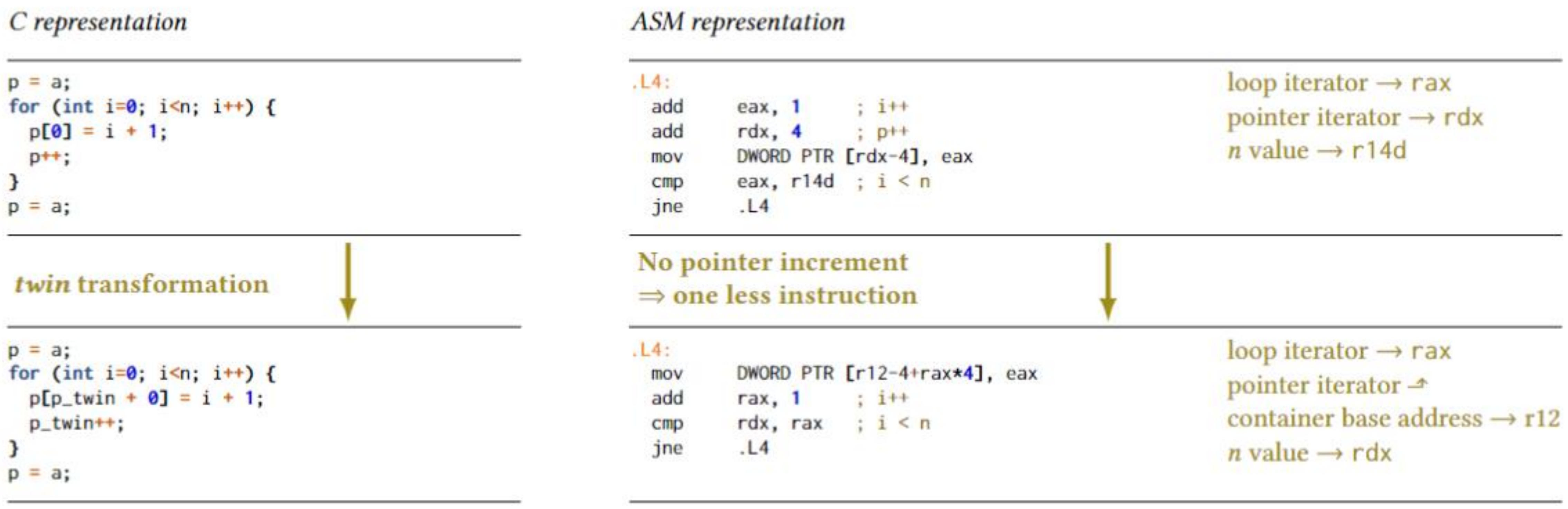
Performance heuristics

- Guide SDFG transformations
- Must handle
  - Application requirements
  - Hardware capabilities

Data instrumentation

- Simplifies debugging
- Allows faster heuristics development

Work on C and pointer analysis



C representation

```
p = a;
for (int i=0; i<n; i++) {
  p[0] = i + 1;
  p++;
}
p = a;
```

ASM representation

```
.L4:
  add     eax, 1      ; i++
  add     rdx, 4      ; p++
  mov     DWORD PTR [rdx-4], eax
  cmp     eax, r14d   ; i < n
  jne     .L4
```

loop iterator ⟶ rax
pointer iterator ⟶ rdx
n value ⟶ r14d

twin **transformation**

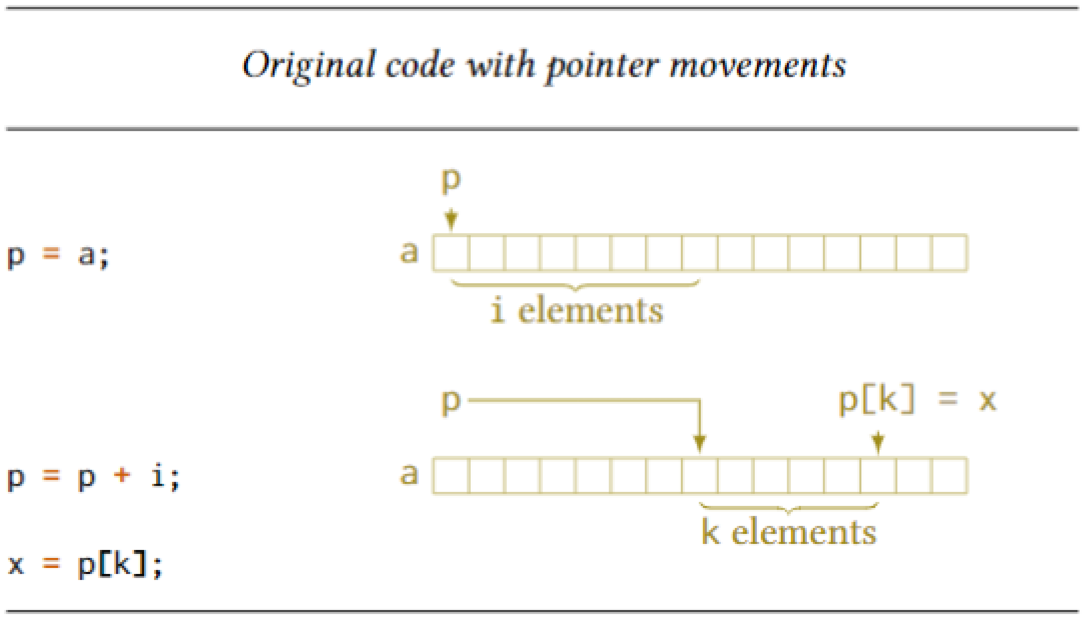
No pointer increment
⇒ one less instruction

```
p = a;
for (int i=0; i<n; i++) {
  p[p_twin + 0] = i + 1;
  p_twin++;
}
p = a;
```

```
.L4:
  mov     DWORD PTR [r12-4+rax*4], eax
  add     rax, 1      ; i++
  cmp     rdx, rax    ; i < n
  jne     .L4
```

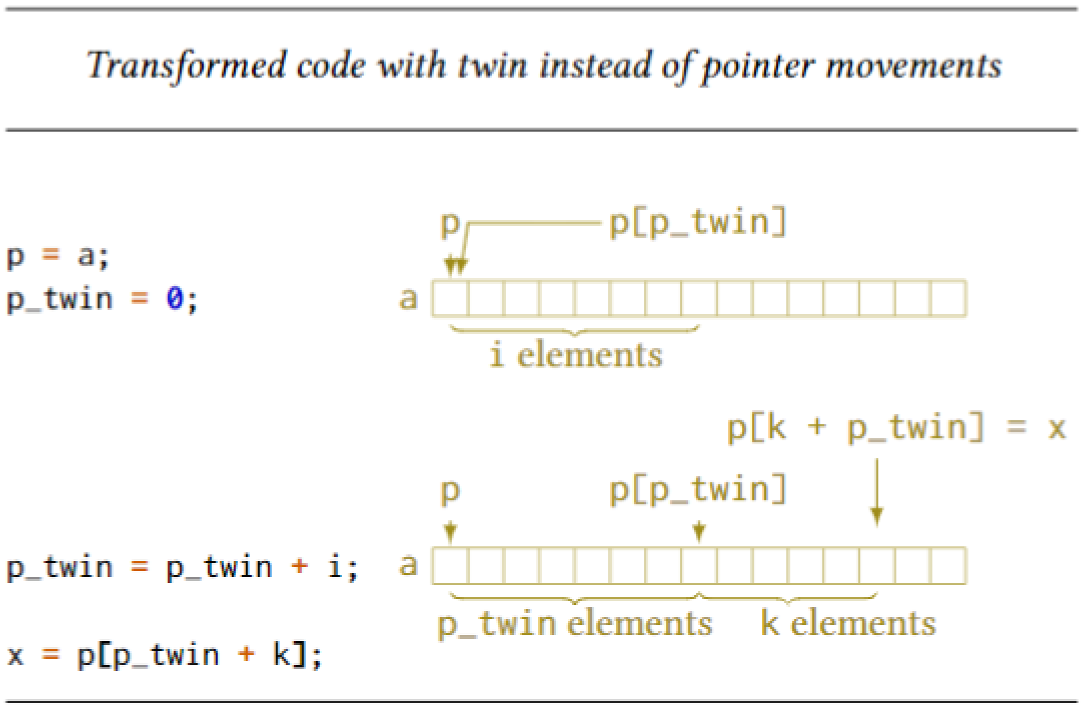
loop iterator ⟶ rax
pointer iterator ⤴
container base address ⟶ r12
n value ⟶ rdx

Work on C and pointer analysis



Original code with pointer movements

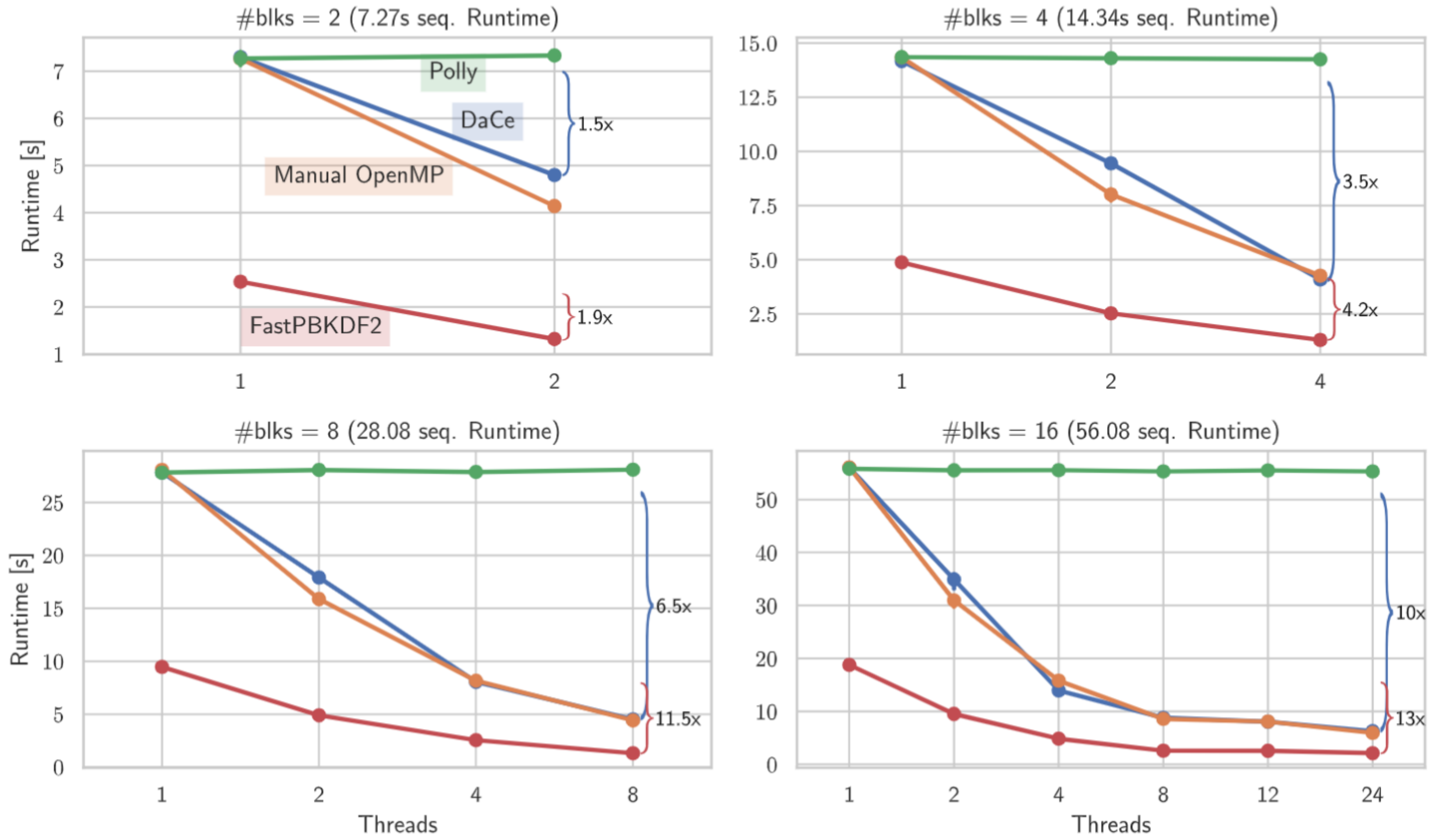
```
p = a;

p = p + i;

x = p[k];
```



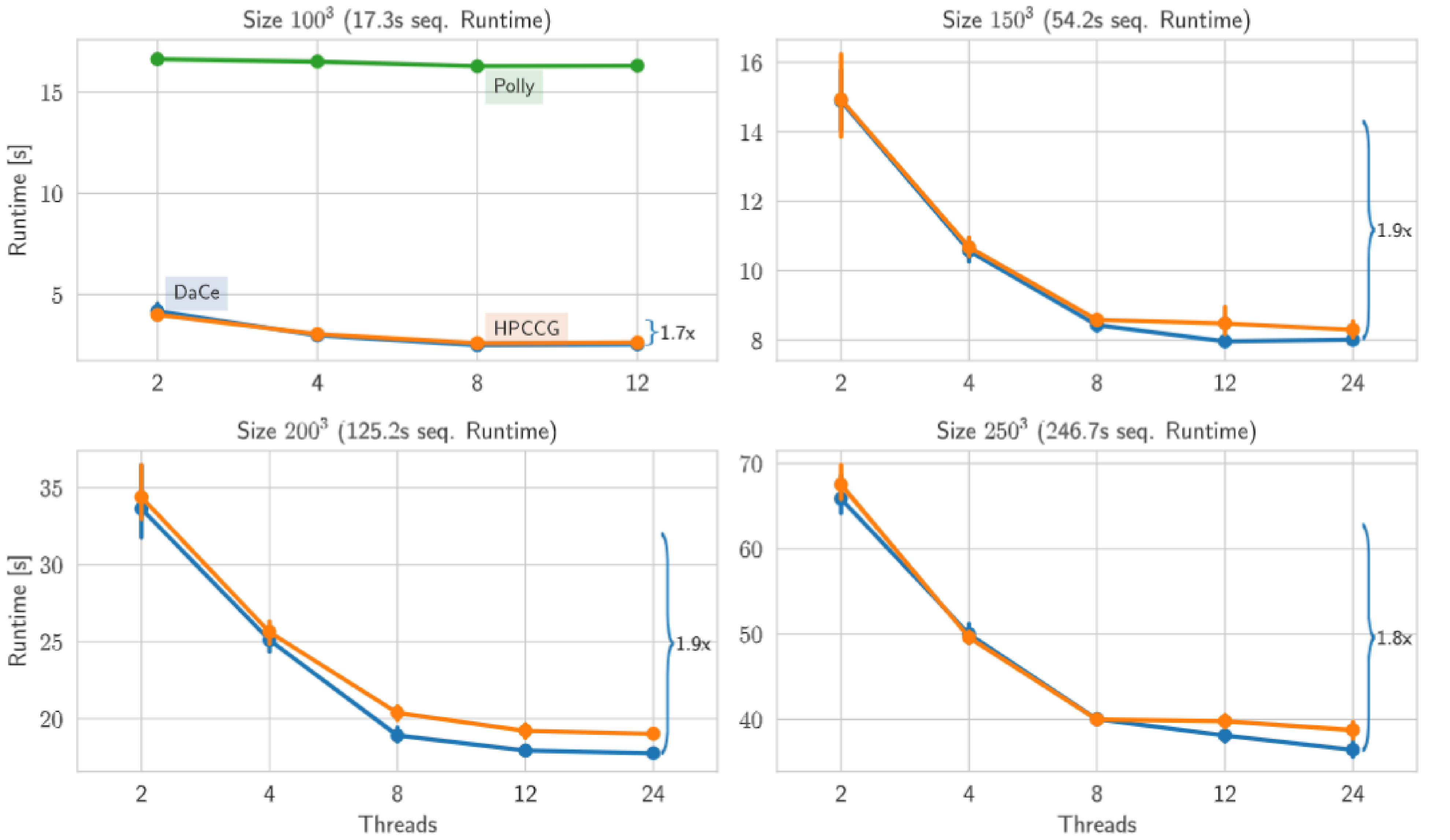
Transformed code with twin instead of pointer movements

```
p = a;
p_twin = 0;

p_twin = p_twin + i;

x = p[p_twin + k];
```

PBKDF2


#blks = 2 (7.27s seq. Runtime)
#blks = 4 (14.34s seq. Runtime)
#blks = 8 (28.08 seq. Runtime)
#blks = 16 (56.08 seq. Runtime)
Runtime [s]
Threads
Polly
DaCe
Manual OpenMP
FastPBKDF2
1.5x
1.9x
3.5x
4.2x
6.5x
11.5x
10x
13x

HPCCG


Size 100^3 (17.3s seq. Runtime)
Size 150^3 (54.2s seq. Runtime)
Size 200^3 (125.2s seq. Runtime)
Size 250^3 (246.7s seq. Runtime)
Runtime [s]
Threads
Polly
DaCe
HPCCG
1.7x
1.9x
1.9x
1.8x
2
4
8
12
24
5
10
15
8
10
12
14
16
20
25
30
35
40
50
60
70