DaFlEx

K-Caching & Array layout

K-Caching

```
REAL  INP1(KLEV)
REAL  OUT1(KLEV)
REAL  TMP1(KLEV)

DO  JK=1,KLEV
    ! f() is some computation
    TMP1(JK) = f(INP1(JK))
END DO

DO  JK=1,KLEV
    OUT1(JK) = TMP1(JK)
END DO
```

Two loops

```
REAL  INP1(KLEV)
REAL  OUT1(KLEV)
REAL  TMP1

DO  JK=1,KLEV
    ! f() is some computation
    TMP1  = f(INP1(JK))
    OUT1(JK) = TMP1
END DO
```

One fused loop

Array layout

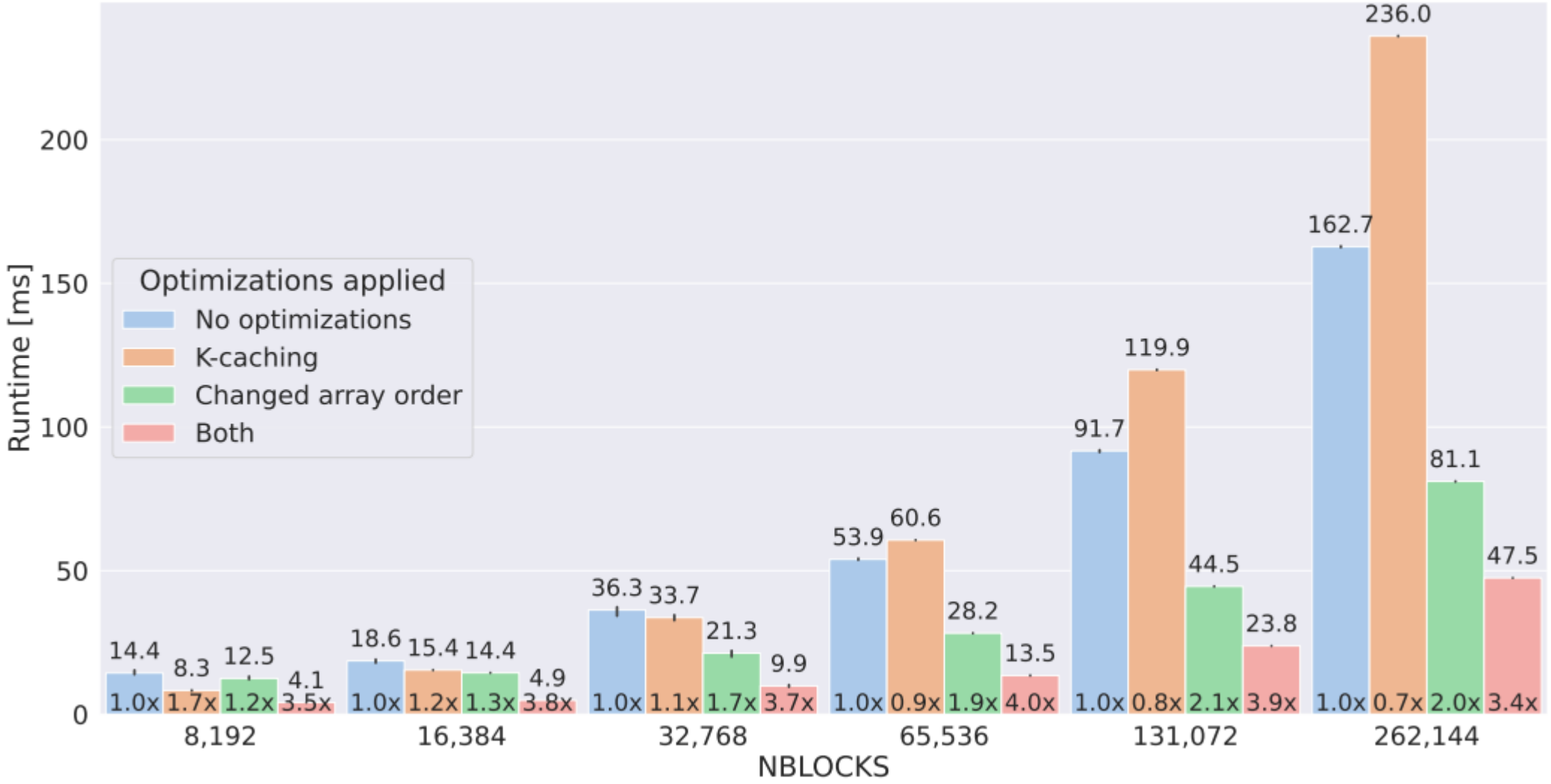
```
SUBROUTINE cloudsc_example(&
        & KLON, KLEV, NBLOCKS, NCLV, NCLDQL, &
        & INP1, INP3, OUT1)

    INTEGER KLON
    INTEGER KLEV
    INTEGER NBLOCKS
    INTEGER NCLV
    INTEGER NCLDQL

    REAL INP1(KLON, KLEV, NBLOCKS)
    REAL INP3(KLON, KLEV, NCLV, NBLOCKS)
    REAL OUT1(KLON, KLEV, NBLOCKS)

    DO JN=1,NBLOCKS
        CALL inner_loops(&
            & KLON, KLEV, NCLV, NCLDQL, &
            & INP1(:, :, JN), INP3(:, :, :, JN), OUT1(:, :, JN))
    ENDDO

END SUBROUTINE cloudsc_example

SUBROUTINE inner_loops(&
        & KLON, KLEV, NCLV, NCLDQL, &
        & INP1, INP3, OUT1)

    INTEGER KLON
    INTEGER KLEV
    INTEGER NCLV
    INTEGER NCLDQL

    REAL INP1(KLON, KLEV)
    REAL INP3(KLON, KLEV, NCLV)
    REAL OUT1(KLON, KLEV)
    REAL TMP(KLON, KLEV)

    DO JK=1,KLEV
        DO JL=1,KLON
            TMP(JL, JK) = INP1(JL, JK) + INP3(JL, JK, 4)
        ENDDO
    ENDDO

    DO JK=1,KLEV
        DO JL=1,KLON
            OUT1(JL, JK) = TMP(JL, JK) - INP3(JL, JK, 4)
        ENDDO
    ENDDO
END SUBROUTINE inner_loops
```

Array layout

JK = 1 JN = 1	JK = 2 JN = 1	JK = 3 JN = 1	JK = 1 JN = 2	JK = 2 JN = 2	JK = 3 JN = 2	JK = 1 JN = 3	JK = 2 JN = 3

JK = 1 JN = 1	JK = 1 JN = 2	JK = 1 JN = 3	JK = 2 JN = 1	JK = 2 JN = 2	JK = 2 JN = 3	JK = 3 JN = 1	JK = 3 JN = 2

Results


Runtime [ms]
0
50
100
150
200
Optimizations applied
No optimizations
K-caching
Changed array order
Both
14.4
8.3
12.5
4.1
1.0x
1.7x
1.2x
3.5x
18.6
15.4
14.4
4.9
1.0x
1.2x
1.3x
3.8x
36.3
33.7
21.3
9.9
1.0x
1.1x
1.7x
3.7x
53.9
60.6
28.2
13.5
1.0x
0.9x
1.9x
4.0x
91.7
119.9
44.5
23.8
1.0x
0.8x
2.1x
3.9x
162.7
236.0
81.1
47.5
1.0x
0.7x
2.0x
3.4x
8,192
16,384
32,768
65,536
131,072
262,144
NBLOCKS