BLAS libraries

for Linear Algebra in Python

BLAS subprograms		Semantic	FP ops.	Mem. ops.
Level 1	Vector Addition	$y_i = x_i + y_i$	n	$3n$
	Vector Scaling	$x_i = sx_i$	n	$2n$
	Dot Product	$s = \sum_{i=0}^{n-1} x_i y_i$	$2n$	$2n$
Level 2	Matrix-vector multiplication	$y_i = y_i + \sum_{j=0} a_{ij} x_j$	$2n^2+n$	n^2+3n
	Rank-one update	$a_{ij} = a_{ij} + x_i y_j$	$2n^2$	$2n^2+2n$
Level 3	Matrix-Matrix multiplication	$c_{ij} = c_{ij} + \sum_{k=0}^{n-1} a_{ik} b_{kj}$	$2n^3+n^2$	$4n^2$

BLAS ROUTINES

matrix multiplication: C = A * B (level 3)

```
DGEMM( TRANSA, TRANSB, M, N, L, ALPHA, A, LDA, B, LDB, BETA, C, LDC )
 'N' or 'T'                                  max(1,M)
```

matrix times vector: Y = A * X (level 2)

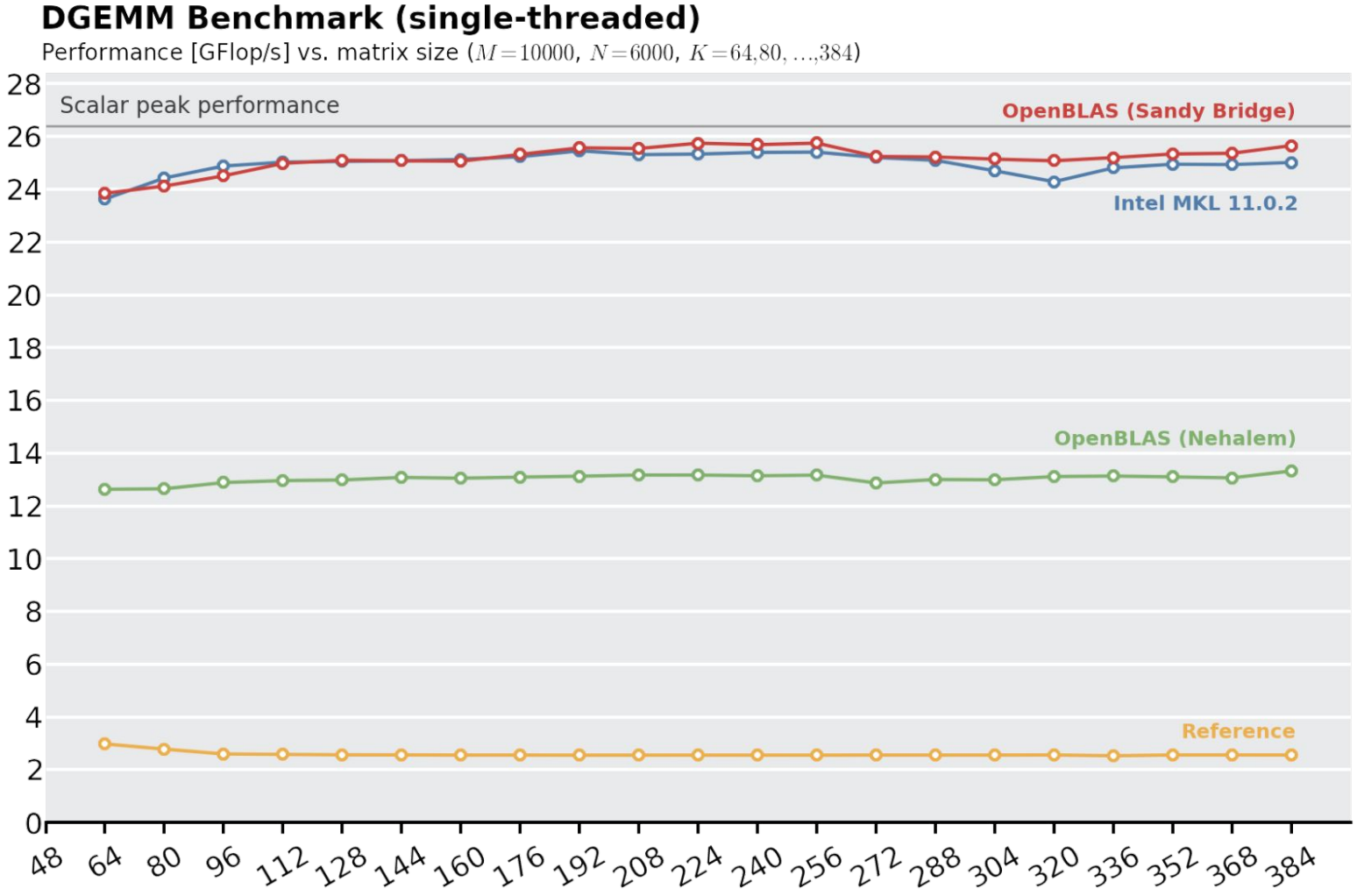
```
DGEMV( TRANS, M, N, ALPHA, A, LDA, X, INCX, BETA, Y, INCY )
               1.0d0                        0.0d0
```

vector swap: X ⇔ Y (level 1)

```
DSWAP( N, X, INCX, Y, INCY )
```

scalar product: p = X'·Y (level 1)

```
p = DDOT( N, X, INCX, Y, INCY )
          Function    Increment for elements
```


DGEMM Benchmark (single-threaded)
Performance [GFlop/s] vs. matrix size ($M=10000$, $N=6000$, $K=64,80,\ldots,384$)
Scalar peak performance
OpenBLAS (Sandy Bridge)
Intel MKL 11.0.2
OpenBLAS (Nehalem)
Reference
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
48
64
80
96
112
128
144
160
176
192
208
224
240
256
272
288
304
320
336
352
368
384