HPC 4MA

VO Van Nghia - PHAM Tuan Kiet

January 24, 2022

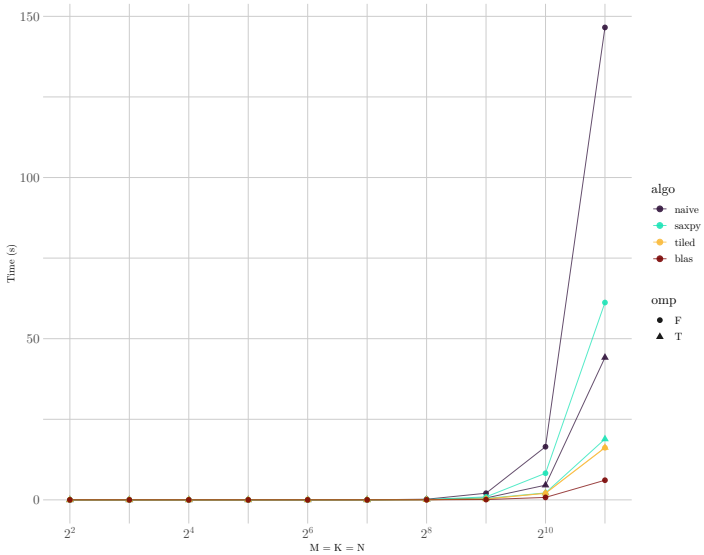
Section 1

Introduction

Section 2

OpenMP

Benchmarking



M and N

```
## ( 1.00 1.50 )
##
## ( 1.00 1.50 )
## ( 1.50 2.00 )
##
## Frobenius Norm = 3.250000
## Total time naive = 0.000000
## Gflops = 0.026229
##
## ( 3.25 0.00 )
##
## Frobenius Norm = 3.250000
## Total time BLAS = 0.019731
## Gflops = 0.000000
##
## ( 3.25 0.00 )
```

```c
double norm(int nrow, int ncol, int ld, double *A);
void print_array(int nrow, int ncol, int ld, double *A);
```

```c
printf("Frobenius Norm = %f\n", norm(N, M, ldc, c));
// ...
print_array(M, N, ldc, c);
```

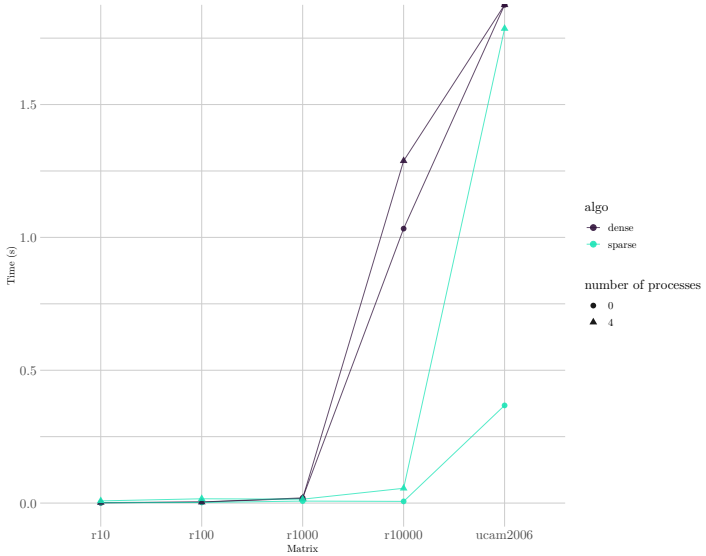
```c
int lda = N + 1;
int ldb = K + 1;
int ldc = N + 1;
double *a = (double *)malloc(lda * K * sizeof(double));
double *b = (double *)malloc(ldb * M * sizeof(double));
double *c = (double *)malloc(ldc * M * sizeof(double));
// ...
cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, M, N, K, alpha, a, lda,
b, ldb, beta, c, ldc);
```

```
int lda = M;
int ldb = K;
int ldc = M;
double *a = (double *)malloc(lda * K * sizeof(double));
double *b = (double *)malloc(ldb * N * sizeof(double));
double *c = (double *)malloc(ldc * N * sizeof(double));
```

Section 3

MPI

Benchmarking



Matrix size is not divisible by number of processes

```
def mpi_all_to_all_allgather(x, xd, comm=MPI.COMM_WORLD):
    comm.Allgather(xd, x)
    return x
```

```
ValueError: message: cannot infer count, number of entries 10 is not a multiple
    of required number of blocks 4
```

```
def mpi_all_to_all_allgatherv(x, xd, counts, disps, comm=MPI.COMM_WORLD):
    comm.Allgatherv([xd, np.size(xd), MPI.DOUBLE], [x, counts, disps, MPI.DOUBLE
    ])
    return x
```

```
    nd = int(np.ceil(n / size))
    start = rank * nd
    end = (rank + 1) * nd
    mpi_all_to_all = mpi_all_to_all_allgather
    if n % size:
        counts = [nd] * size
        counts[-1] = n - nd * (size - 1)
        disps = np.zeros(size, dtype=int)
        disps[1:] = np.cumsum(counts)[:-1]
        mpi_all_to_all = functools.partial(
            mpi_all_to_all_allgatherv, counts=counts, disps=disps
        )
    if rank == size - 1:
        end = n
        nd = end - start
```

Thank you for your attention !