

Tools/likwid/example_marker_api_dgemm

Example: likwid Marker API in dgemm

- Tutorial Marker API with C/C++
- Add likwid marker API to dgemm source code

```
cp -av dgemm.multithread{,.likwid}.c
vim dgemm.multithread.likwid.c
diff -u dgemm.multithread{,.likwid}.c

--- dgemm.multithread.c 2020-02-17 09:42:24.902159044 +0100
+++ dgemm.multithread.likwid.c 2023-05-30 13:42:28.067665741 +0200
@@ -28,6 +28,9 @@
#include "stats.h"
#include "timing.h"

+// include header file of likwid API
+#include "likwid.h"
+
+// Add compiler hint: no pointer aliasing
+// See: https://en.wikipedia.org/wiki/Restrict
+#define DGEMM_RESTRICT __restrict__
@@ -92,6 +95,15 @@
printf("Number of repetitions set to %i. Overwrite with command line option -m
}

+ // initialize likwid marker API
+ likwid_markerInit();
+ #pragma omp parallel
+ {
+ likwid_markerThreadInit();
+ likwid_markerRegisterRegion( "cblas_dgemm" );
+ likwid_markerRegisterRegion( "validate" );
+ }
+
+ int divisor = 1;
+ #if defined(USE_UNROLLED_4X) || defined(USE_UNROLLED_4X_SIMD_4X)
```

```

        divisor = 4;
@@ -159,8 +171,14 @@
        for (int r = 0; r < repeats; r++) {
            const double start = get_time_monotonic();
            #if defined(USE_MKL) || defined(USE_CBLAS)
+           #pragma omp parallel
+           likwid_markerStartRegion( "cblas_dgemm" );
+
            cblas_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans,
                        N, N, N, alpha, matrixA, N, matrixB, N, beta, matrixC, N);
+
+           #pragma omp parallel
+           likwid_markerStopRegion( "cblas_dgemm" );
            #elif defined(USE_NVBLAS)
                char transA = 'N';
                char transB = 'N';
@@ -524,6 +542,8 @@
                gigaFlops[r] = (flops_per_step / time_taken) / 1000000000.0;
            }

+           #pragma omp parallel
+           likwid_markerStartRegion( "validate" );
            double infNorm =
                matrix_check(
                    N,
@@ -531,6 +551,9 @@
                    alpha, beta,
                    repeats);

+           #pragma omp parallel
+           likwid_markerStopRegion( "validate" );
+
            printf("\n");
            printf("=====\n");

@@ -577,5 +600,8 @@
            free(matrixB);
            free(matrixC);

+           // Close likwid marker API
+           likwid_markerClose();
+
            return 0;
        }

```

- Prepare environment

```
module purge
module add \
    compiler/gnu/12 \
    numlib/mkl/2022
```

- Build dgemm benchmark with likwid marker API

```
gcc -std=c11 -Ofast -march=native -flto -fopenmp \
    -DUSE_MKL \
    -DLIKWID_PERFMON \
    timing.c stats.c matrix_common.c dgemm.multithread.likwid.c -o dgemm \
    -L${MKLROOT}/lib/intel64 -Wl,--no-as-needed -lmkl_intel_lp64 -lmkl_gnu_thread -lmkl
    -lm -llikwid
```

- Run benchmark with 76 threads

```
# Run with 76 threads
export OMP_NUM_THREADS=76
export MKL_NUM_THREADS=76
```

```
likwid-perfctr --marker --group FLOPS_AVX -C 0-75 \
    ./dgemm -m 30 -n 8000
```

```
-----
CPU name:      Intel(R) Xeon(R) Platinum 8368 CPU @ 2.40GHz
CPU type:      Intel Icelake SP processor
CPU clock:     2.39 GHz
-----
```

```
Matrix size: 8000
Repeat multiply 30 times.
Alpha =      1.000000
Beta  =      1.000000
Allocating Matrices...
Allocation complete, populating with values...
Performing multiplication...
Calculating matrix check...
```

```
=====
|| E ||_∞:      0.000000E+00
-> Solution check PASSED successfully.
Memory for Matrices: 1464.843750 MB
Multiply time:      6.905394 seconds
FLOPs computed:     30723840000000.000000
Min GFLOP/s:        4188.570941 GF/s
Max GFLOP/s:        4657.151969 GF/s
Average GFLOP/s:    4453.455421 GF/s
Std. dev. GFLOP/s:  743.452370 GF/s
Median GFLOP/s:     4482.623427 GF/s
```

MAD GFLOP/s: 91.773075 GF/s

Region cblas_dgemm, Group 1: FLOPS_AVX

...					
Event		Counter	Sum	Min	
...					
FP_ARITH_INST_RETIRED_256B_PACKED_SINGLE STAT		PMC0	0	0	
FP_ARITH_INST_RETIRED_256B_PACKED_DOUBLE STAT		PMC1	0	0	
FP_ARITH_INST_RETIRED_512B_PACKED_SINGLE STAT		PMC2	0	0	
FP_ARITH_INST_RETIRED_512B_PACKED_DOUBLE STAT		PMC3	3845039980000	49742620000	
...					
Metric		Sum	Min	Max	Avg
...					
Packed DP [MFLOP/s] STAT		4.489540e+06	58084.5401	59718.1969	59072.8959

Region validate, Group 1: FLOPS_AVX

...					
Event		Counter	Sum	Min	
...					
FP_ARITH_INST_RETIRED_256B_PACKED_SINGLE STAT		PMC0	0	0	
FP_ARITH_INST_RETIRED_256B_PACKED_DOUBLE STAT		PMC1	32000000	420000	4
FP_ARITH_INST_RETIRED_512B_PACKED_SINGLE STAT		PMC2	0	0	
FP_ARITH_INST_RETIRED_512B_PACKED_DOUBLE STAT		PMC3	0	0	
...					
Metric		Sum	Min	Max	Avg
...					
Packed DP [MFLOP/s] STAT		48822.1970	603.0999	664.1347	642.3973

- Region cblas_dgemm uses only AVX512 operations
- Region validate uses only AVX2 operations
- Computed: Average GFLOP/s: 4453.455421 GF/s

- Measured: Packed DP [MFLOP/s] STAT: 4.489540e+06 == 4489.540 GF/s