Parallelware Training Series

Motif-guided Parallelization of ZPIC with OpenMP and OpenACC

Office hours

Homework exercises demo

Manuel Arenaz | Oct-Nov 2020

NERSC

appentra
make code parallel

OpenACC
More Science, Less Programming

OpenMP
Enabling HPC since 1997

Agenda

www.nersc.gov/users/training/events/parallelware-training-series-oct-nov-2020/

Part 1, Tuesday, October 27

8:30 am - 12:00 pm PDT

Introduction to Parallelware tools: Ensuring parallel programming best practices

- 10' - Introduction by NERSC
- 10' - Welcome
- 20' - Introduction
- 5' break
- 20' - NESAP Applications & Motifs
- 30' - Catalog of defects and best practice recommendations
- 5' break
- 60' - Parallelware tools: Trainer & Analyzer
- 5' break
- 20' - Homework
- 20' - Q&A

Format: Remote lectures, and demos, and homework exercises

Part 2, Thursday, October 29

9:00 am - 12:00 am PDT

Office hours [Optional]

- **60' - Homework exercises demo**
- 5' break
- 110' - Support, Questions, FAQs for using Parallelware tools

Format: Remote office hours

Part 3, Wednesday, November 4

8:00 am - 1:00 pm PST

Guided parallelization of ZPIC: Ensuring best practices with Parallelware tools

- 30' - Case study: Guided parallelization of ZPIC with Parallelware tools
- 20' - Performance evaluation of ZPIC
- 10' break
- 220' - Bring your own applications
- 20' - Q&A

Format: Remote demos and hands-on

ATMUX

- Sparse linear algebra
- Multiplication of Transposed Sparse Matrix by Vector

$$y = A^T x = \begin{pmatrix} 0 & 40 & 0 & 0 & 0 \\ 0 & 55 & 56 & 9 & 34 \\ 18 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 57 \\ 0 & 21 & 0 & 29 & 0 \end{pmatrix}^T \cdot \begin{pmatrix} 83 \\ 86 \\ 77 \\ 15 \\ 93 \end{pmatrix} = \begin{pmatrix} 18 \times 77 \\ 40 \times 83 + 55 \times 86 + 21 \times 93 \\ 56 \times 86 \\ 9 \times 86 + 29 \times 93 \\ 34 \times 86 + 57 \times 15 \end{pmatrix} = \begin{pmatrix} 1386 \\ 10003 \\ 4816 \\ 3471 \\ 3779 \end{pmatrix}$$

- Using an sparse/compressed storage format: Compressed Row Storage (CRS)

$$A = \begin{array}{c} \begin{matrix} c0 & c1 & c2 & c3 & c4 \end{matrix} \\ \begin{pmatrix} 0 & 40 & 0 & 0 & 0 \\ 0 & 55 & 56 & 9 & 34 \\ 18 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 57 \\ 0 & 21 & 0 & 29 & 0 \end{pmatrix} \begin{matrix} r0 \\ r1 \\ r2 \\ r3 \\ r4 \end{matrix} \end{array}$$

val[9] = { 40, 55, 56, 9, 34, 18, 57, 21, 29 }

row_ptr[6] = { 0, 1, 5, 6, 7, 9 }

col_ind[9] = { 1, 1, 2, 3, 4, 0, 4, 1, 3 }

Profiling of ATMUX

```
$ gcc atmux.c lib/*.c -I lib -pg -o atmux
$ ./atmux 17000
$ gprof ./atmux
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self   total
 time   seconds   seconds  calls   s/call   s/call  name
 38.59       4.80   4.80         1     4.80   4.80  CRSMatrix_from
 31.95       8.78   3.98         1     3.98   3.98  Matrix2D_randSparse
 29.53      12.46  3.68         10     0.37   0.37  compute
  0.00      12.46  0.00         10     0.00   0.37  atmux
  0.00      12.46  0.00         2      0.00   0.00  Vector_delete
  0.00      12.46  0.00         2      0.00   0.00  Vector_new
  0.00      12.46  0.00         2      0.00   0.00  getClock
  0.00      12.46  0.00         1      0.00   0.00  CRSMatrix_delete
  0.00      12.46  0.00         1      0.00   0.00  CRSMatrix_new
  0.00      12.46  0.00         1      0.00   0.00  Matrix2D_delete
  0.00      12.46  0.00         1      0.00   0.00  Matrix2D_new
  0.00      12.46  0.00         1      0.00   0.00  Vector_checksum
  0.00      12.46  0.00         1      0.00   0.00  Vector_rand
```

How to verify correctness of ATMUX

```
$ ./atmux 17000
- Input parameters
size      = 17000
- Executing test...
time (s) = 3.668223
size      = 17000
sparsity = 0.66
chksum    = 244110871193
iters     = 10
```

```
void compute(double *val, double *x, double *y, int *col_ind, int *row_ptr, int n, long long nnz) {
  // y = A^T x
  for (int i = 0; i < n; i++) {
        for (int k = row_ptr[i]; k < row_ptr[i + 1]; k++) {
            y[col_ind[k]] = y[col_ind[k]] + x[i] * val[k];
        }
  }
}

// Compute sparse matrix-vector multiplication
void atmux(CRSMatrix *in_sparseMat, Vector *in_vec, Vector *out_vec, int n) {

        double *val = CRSMatrix_getData(in_sparseMat);
        double *x = Vector_getData(in_vec);
        double *y = Vector_getData(out_vec);
        int *col_ind = CRSMatrix_colRef(in_sparseMat);
        int *row_ptr = CRSMatrix_rowRef(in_sparseMat);
        long long nnz = CRSMatrix_getSize(in_sparseMat);

        int t, i, k;

        for (t = 0; t < n; t++)
            y[t] = 0;

        compute(val, x, y, col_ind, row_ptr, n, nnz);
}
```

ATMUX: Results & Performance on NERSC Cori

Version	Problem size	#threads	Time (sec) [Speedup]	checksum
atmux_gcc	Size 17000	n/a	1.15	244110871193
atmux_cpu_omp_atomic_gcc		4	[0.36x] 3.53	244110871193
atmux_cpu_omp_explicit_gcc		4	[2.40x] 0.48	244110871193
atmux_gpu_omp_atomic_gcc		n/a	[0.35x] 3.24	244110871193
atmux_gpu_acc_atomic_gcc		n/a	[0.33x] 3.44	244110871193
atmux_gpu_acc_atomic_pgi		n/a	[0.56x] 2.06	244110871193

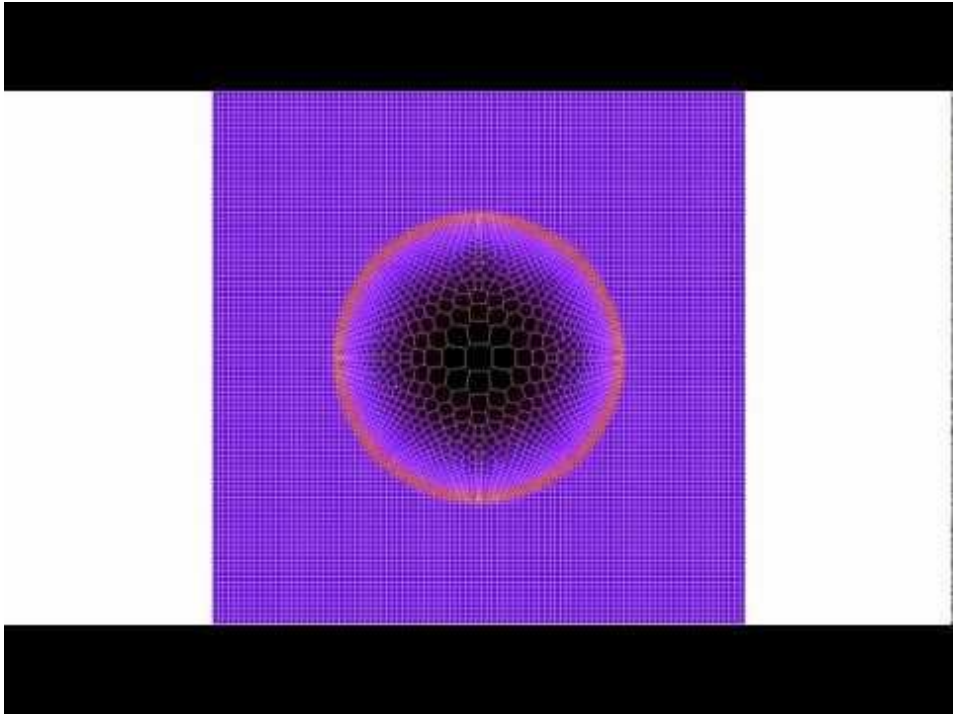
CPU: 2 x Intel Xeon Gold 6148 ('Skylake') @ 2.40 GHz (4 threads allocated)
MEMORY: 384 GB DDR4 memory
GPU: 8 x NVIDIA V100, each with with 16 GB HBM2 memory, connected with NVLink interconnect (1 allocated)

CORAL Benchmarks: LULESH

Livermore **U**nstructured **L**agrange
Explicit **S**hock **H**ydrodynamics

Part of a Physics Simulation software (ALE3D)

Models the propagation of a Sedov blast wave using Lagrangian hydrodynamics



Profiling of LULESH microkernel

```
$ gcc -pg -o luleshmk luleshmk.c -lm
$ ./luleshmk
$ gprof ./luleshmk
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 56.23     24.62    24.62 223680000     0.00     0.00  CalcElemFBHourglassForce_workload
 22.76     34.59     9.97      932     0.01     0.01  ApplyMaterialPropertiesForElems_workload
 15.26     41.27     6.68 27960000     0.00     0.00  CalcElemVelocityGradient_workload
  2.26     42.26     0.99      932     0.00     0.03  CalcFBHourglassForceForElems
  2.24     43.24     0.98 27960000     0.00     0.00  CalcElemFBHourglassForce
  0.75     43.57     0.33 27960000     0.00     0.00  CalcElemVelocityGradient
  0.34     43.72     0.15        1     0.15    43.76  luleshmk
  0.09     43.76     0.04      932     0.00     0.01  CalcKinematicsForElems
  0.09     43.80     0.04                             frame_dummy
  0.00     43.80     0.00      932     0.00     0.01  ApplyMaterialPropertiesForElems
  0.00     43.80     0.00        2     0.00     0.00  calculate_checksum
  0.00     43.80     0.00        2     0.00     0.00  getClock
  0.00     43.80     0.00        1     0.00     0.00  Parameters_create
  0.00     43.80     0.00        1     0.00     0.00  Parameters_free
  0.00     43.80     0.00        1     0.00     0.00  VerifyAndWriteFinalOutput
```

How to verify correctness of LULESH

```
$ ./luleshmk
- Configuring the test...
- Executing the test...
gprof ./luleshmk
- Verifying the test...
Run completed:
   Problem size        =  30
   MPI tasks           =  1
   Iteration count     =  932
   Final Origin Energy = 1.000000e+00
   Number of nodes    = 27000
   Number of elements = 30000
   Number of regions  = 1
      Region 1 of size 30000
   Testing Plane 0 of Energy Array on rank 0:
        MaxAbsDiff   = 8.410000e+02
        TotalAbsDiff = 1.303550e+05
        MaxRelDiff   = 9.655568e-01

Elapsed time         =      71.00 (s)
Grind time (us/z/c)  =   2.821491 (per dom)  (  2.821491
overall)
FOM                  =  354.42254 (z/s)
```

LULESHmk: Results & Performance on NERSC Cori

Version	Problem size	#threads	Time (sec) [Speedup]	checksum_f	checksum_e
luleshmk	NumNodes 27000 NumElems 30000	n/a	70.78	3.28901e+11	4.37594e+12
luleshmk_v1_omp_atomic		4	[1.80x] 39.37	3.28901e+11	4.37594e+12
luleshmk_v2_omp_explicit		4	[1.77x] 40.09	3.28901e+11	4.37594e+12
luleshmk_v3_omp		4	[3.75x] 18.87	3.28901e+11	4.37594e+12
luleshmk_v4_acc_atomic_gcc		n/a	[2.39x] 29.62	3.28901e+11	4.37594e+12
luleshmk_v4_acc_atomic_pgi		n/a	[24.49x] 2.89	3.28901e+11	4.37594e+12
luleshmk_v6_acc_omp_gcc		4	[2.38x] 29.68	3.28901e+11	4.37594e+12
luleshmk_v6_acc_omp_pgi		4	[24.40x] 2.90	3.28901e+11	4.37594e+12

CPU: 2 x Intel Xeon Gold 6148 ('Skylake') @ 2.40 GHz (16 threads allocated)
MEMORY: 384 GB DDR4 memory (32 GB allocated)
GPU: 8 x NVIDIA V100, each with with 16 GB HBM2 memory, connected with NVLink interconnect (2 allocated)

Parallelware Training Series

Motif-guided Parallelization of ZPIC with OpenMP and OpenACC

Office hours

Homework exercises demo