

Scaling behavior of psana1 - Part 2 - test with command perf stat

Content

- [Content](#)
- [2024-02-06 Test of milano216 host with perf](#)
 - [Description](#)
 - [Results](#)
 - [Summary](#)
- [2024-02-07 Test of milano216 host with command perf](#)
 - [Description](#)
 - [Results](#)
 - [Summary](#)
- [2024-02-09 Test of milano216 host with command perf](#)
 - [Description](#)
 - [Summary](#)
- [References](#)

previous page: [Scaling behavior of psana1 - Part 1 - det.calib method in multicore processing with mpi](#)

2024-02-06 Test of milano216 host with perf

Description

Using command:

perf stat -e cache-references,cache-misses,cycles,instructions,branches,branch-misses,faults,migrations,page-faults,L1-dcache-load-misses,L1-icache-load-misses python test-scaling-subproc.py <parameter>

where parameter defines test for different number of CPUs, e.g. <parameter> = 1,2,8,13,16,17,18 stands for test on single, 8, 16, 32, 56, 64, 128 CPUs.

Code of time_consuming_algorithm

```
import numpy as np
from time import time, sleep

def random_standard(shape=(40,60), mu=200, sigma=25, dtype=np.float64):
    a = mu + sigma*np.random.standard_normal(shape)
    return np.require(a, dtype)

def random_arrays(sh2d = (8*512,1024), dtype=np.float64):
    sh3d = (3,) + sh2d
    return random_standard(shape=sh2d, mu=10, sigma=2, dtype=dtype),\
        random_standard(shape=sh3d, mu=20, sigma=3, dtype=dtype)

def time_consuming_algorithm():
    t01 = time()
    a, b = random_arrays()
    t02 = time()
    gr1 = a>=11
    gr2 = (a>9) & (a<11)
    gr3 = a<=9
    t03 = time()
    a[gr1] -= b[0, gr1]
    a[gr2] -= b[1, gr2]
    a[gr3] -= b[2, gr3]
    t04 = time()
    return (t01, t02, t03, t04)
```

Code of the event loop in test-scaling-subproc.py

```
def do_algo(cpu=0, cmt='v0'):  
  
    hostname = get_hostname()  
    #cpu_num = psutil.Process().cpu_num()  
    print('requested cpu:%03d' % cpu)  
  
    SAVE_FIGS = True  
    SHOW_FIGS = False  
    nevents = 100  
    ntpoints = 6  
    arrts = np.zeros((nevents, ntpoints), dtype=np.float64)  
    t05_old = time()  
  
    for nevt in range(nevents):  
        t00 = time()  
        times = time_consuming_algorithm()  
        cpu_num = psutil.Process().cpu_num()  
        #if cpu_num >=16 and cpu_num <=23:  
        #    print('cpu_num:%03d nevt:%03d time:%.6f CPU_NUM IN WEKA RANGE [16,23]' % (cpu_num, nevt, dt_sec))  
        t05 = time()  
        times = (t00,) + times + (t05,)  
        arrts[nevt,:] = times  
        dt_evt = t05 - t05_old  
        t05_old = t05  
        if nevt%10>0: continue  
        dt_alg = times[4] - times[3]  
        dt_in = times[4] - times[1]  
        print('cpu_num:%03d nevt:%03d times (sec)' % (cpu_num, nevt), \  
            ' random arrts: %.6f' % (times[2] - times[1]), \  
            ' indeces: %.6f' % (times[3] - times[2]), \  
            ' alg: %.6f' % (times[4] - times[3]), \  
            ' inside algo: %.6f' % (times[4] - times[1]), \  
            ' per event: %.6f' % dt_evt)  
    ...  
    further code is ffor saving results and graphics
```

Results

Results of the test for 1,8,16,32,56,64,128 CPU

ana-4.0.59-py3 [dubrovin@sdfmilan216:~/LCLS/con-py3]\$

1 CPU

=====

Performance counter stats for 'python test-scaling-subproc.py -1':

4,522,410,200	cache-references:u			(62.49%)
112,207,635	cache-misses:u	#	2.481 % of all cache refs	(62.51%)
224,402,878,245	cycles:u			(62.51%)
428,582,543,872	instructions:u	#	1.91 insn per cycle	(62.51%)
59,430,436,824	branches:u			(62.50%)
2,353,206,592	branch-misses:u	#	3.96% of all branches	(62.50%)
657,277	faults:u			
0	migrations:u			
657,277	page-faults:u			
2,169,783,808	L1-dcache-load-misses:u			(62.50%)
7,173,374	L1-icache-load-misses:u			(62.50%)

70.762930452 seconds time elapsed

66.918003000 seconds user

2.380196000 seconds sys

8 CPU

=====

Performance counter stats for 'python test-scaling-subproc.py -2':

35,293,654,947	cache-references:u			(62.50%)
675,772,563	cache-misses:u	#	1.915 % of all cache refs	(62.50%)
1,863,835,416,629	cycles:u			(62.50%)
3,408,694,078,315	instructions:u	#	1.83 insn per cycle	(62.50%)
470,729,321,611	branches:u			(62.50%)
18,710,029,709	branch-misses:u	#	3.97% of all branches	(62.50%)
4,759,204	faults:u			
0	migrations:u			
4,759,204	page-faults:u			
17,164,781,068	L1-dcache-load-misses:u			(62.50%)
42,407,266	L1-icache-load-misses:u			(62.50%)

82.107165073 seconds time elapsed

600.726489000 seconds user

28.169314000 seconds sys

16 CPU

=====

Performance counter stats for 'python test-scaling-subproc.py -8':

71,125,012,043	cache-references:u			(62.50%)
2,509,743,885	cache-misses:u	#	3.529 % of all cache refs	(62.50%)
4,256,512,072,612	cycles:u			(62.50%)
6,815,210,853,848	instructions:u	#	1.60 insn per cycle	(62.50%)
940,797,592,651	branches:u			(62.50%)
37,401,077,277	branch-misses:u	#	3.98% of all branches	(62.50%)
9,874,603	faults:u			
0	migrations:u			
9,874,603	page-faults:u			
34,764,585,133	L1-dcache-load-misses:u			(62.50%)
82,908,203	L1-icache-load-misses:u			(62.50%)

98.180409648 seconds time elapsed

1370.175346000 seconds user

121.864448000 seconds sys

32 CPU

=====

Performance counter stats for 'python test-scaling-subproc.py -13':

140,229,421,945	cache-references:u			(62.50%)
5,022,345,750	cache-misses:u	#	3.582 % of all cache refs	(62.50%)
8,558,410,936,114	cycles:u			(62.50%)
13,628,360,184,584	instructions:u	#	1.59 insn per cycle	(62.50%)
1,881,291,550,548	branches:u			(62.50%)
74,783,808,615	branch-misses:u	#	3.98% of all branches	(62.50%)
19,579,143	faults:u			
0	migrations:u			
19,579,143	page-faults:u			
68,615,480,748	L1-dcache-load-misses:u			(62.50%)
163,094,161	L1-icache-load-misses:u			(62.50%)

99.279801084 seconds time elapsed

2763.979749000 seconds user

246.852789000 seconds sys

56 CPU

=====

Performance counter stats for 'python test-scaling-subproc.py -16':

245,664,589,385	cache-references:u			(62.50%)
5,986,128,102	cache-misses:u	#	2.437 % of all cache refs	(62.50%)
13,462,198,820,573	cycles:u			(62.50%)
23,847,765,747,744	instructions:u	#	1.77 insn per cycle	(62.50%)

```

3,290,927,488,525    branches:u                                (62.50%)
130,897,170,304     branch-misses:u                #    3.98% of all branches    (62.50%)
 35,494,247         faults:u
      0             migrations:u
 35,494,247         page-faults:u
119,933,873,577     L1-dcache-load-misses:u        (62.50%)
 288,403,921       L1-icache-load-misses:u        (62.50%)

```

108.453630713 seconds time elapsed

5381.177612000 seconds user

333.903330000 seconds sys

64 CPU

=====

Performance counter stats for 'python test-scaling-subproc.py -17':

```

281,639,175,978    cache-references:u                                (62.50%)
 8,968,404,974     cache-misses:u                #    3.184 % of all cache refs    (62.50%)
16,140,364,752,053 cycles:u                                (62.50%)
27,256,133,511,829 instructions:u                #    1.69  insn per cycle        (62.50%)
 3,761,710,111,186 branches:u                                (62.50%)
149,569,155,086    branch-misses:u                #    3.98% of all branches    (62.50%)
 39,148,442        faults:u
      0             migrations:u
 39,148,442        page-faults:u
137,584,278,754    L1-dcache-load-misses:u        (62.50%)
 330,750,296       L1-icache-load-misses:u        (62.50%)

```

120.688547006 seconds time elapsed

6274.688233000 seconds user

484.406164000 seconds sys

120 CPU

=====

Performance counter stats for 'python test-scaling-subproc.py -18':

```

532,229,037,371    cache-references:u                                (62.50%)
14,227,944,434     cache-misses:u                #    2.673 % of all cache refs    (62.50%)
29,404,359,241,173 cycles:u                                (62.50%)
51,095,884,028,391 instructions:u                #    1.74  insn per cycle        (62.50%)
 7,053,547,766,317 branches:u                                (62.50%)
280,479,284,507    branch-misses:u                #    3.98% of all branches    (62.50%)
 73,250,012        faults:u
      0             migrations:u
 73,250,012        page-faults:u
260,078,672,869    L1-dcache-load-misses:u        (62.50%)
 618,858,635       L1-icache-load-misses:u        (62.50%)

```

119.736692035 seconds time elapsed

11628.275939000 seconds user

843.423292000 seconds sys

Summary

number of CPU	cache- references	cache- misses	cycles	instructions	branches	branch- misses	faults	page- faults	L1- dcache- load- misses	L1-icache- load- misses	L1- icache Ratio N /1	cmt
1	4,522,410,200	112,207,635	224,402,878,245	428,582,543,872	59,430,436,824	2,353,206,592	657,277	657,277	2,169,783,808	7,173,374	1	
8	35,293,654,947	675,772,563				18,710,029,709			17,164,781,068	42,407,266	5.9	
16	71,125,012,043	2,509,743,885				37,401,077,277			34,764,585,133	82,908,203	11.6	
32	140, 229,421,945	5,022,345,750				74,783,808,615			68,615,480,748	163,094,161	22.7	
56	245, 664,589,385	5,986,128,102				130, 897,170,304			119, 933,873,577	288,403,921	40.2	

64	281, 639,175,978	8,968,404,974				149, 569,155,086			137, 584,278,754	330,750,296	46.1	
120	532, 229,037,371	14,227, 944,434	29,404,359,241,1 73	51,095, 884,028,391	7,053, 547,766,317	280, 479,284,507	73,250, 012	73,250,012	260, 078,672,869	618,858,635	86.2	

2024-02-07 Test of milano216 host with command perf

Description

Running perf with mpirun on a single and 80 CPUs:

perf stat -e cache-references,cache-misses,cycles,instructions,branches,branch-misses,faults,migrations,page-faults,L1-dcache-load-misses,L1-icache-load-misses,dTLB-load-misses,iTLB-load-misses mpirun -n 1 python Detector/examples/test-scaling-mpi.py

perf stat -e cache-references,cache-misses,cycles,instructions,branches,branch-misses,faults,migrations,page-faults,L1-dcache-load-misses,L1-icache-load-misses,dTLB-load-misses,iTLB-load-misses mpirun -n 80 python Detector/examples/test-scaling-mpi.py

Code of the time_consuming_algorithm

```
import numpy as np
from time import time

def random_standard(shape=(40,60), mu=200, sigma=25, dtype=np.float64):
    a = mu + sigma*np.random.standard_normal(shape)
    return np.require(a, dtype)

def random_arrays(sh2d = (8*512,1024), dtype=np.float64):
    sh3d = (3,) + sh2d
    return random_standard(shape=sh2d, mu=10, sigma=2, dtype=dtype),\
        random_standard(shape=sh3d, mu=20, sigma=3, dtype=dtype)

def time_consuming_algorithm():
    a, b = random_arrays()
    gr1 = a>=11
    gr2 = (a>9) & (a<11)
    gr3 = a<=9
    t0_sec = time()
    a[gr1] -= b[0, gr1]
    a[gr2] -= b[1, gr2]
    a[gr3] -= b[2, gr3]
    return time() - t0_sec
```

Code of the event loop in Detector/examples/test-scaling-mpi.py

```
from mpi4py import MPI
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

hostname = get_hostname()
cpu_num = psutil.Process().cpu_num()
print('rank:%02d cpu_num:%03d size:%02d' % (rank, cpu_num, size))

ranks = (0, 10, 20, 30, 40, 50, 60, 70)
SAVE_FIGS = True
SHOW_FIGS = False
nevents = 100
arrts = np.zeros((nevents, size), dtype=np.float64)

for nevt in range(nevents):
    dt_sec = time_consuming_algorithm()
    arrts[nevt,rank] = dt_sec # dt_sec = time()-t0_sec
    cpu_num = psutil.Process().cpu_num()
    if cpu_num >=16 and cpu_num <=23:
        print('rank:%02d cpu_num:%03d nevt:%03d time:%.6f CPU_NUM IN WEKA RANGE [16,23]' % (rank, cpu_num,
nevt, dt_sec))
        if nevt%10>0: continue
        print('rank:%02d cpu_num:%03d nevt:%03d time:%.6f' % (rank, cpu_num, nevt, dt_sec))
    ...
somme graphics for array arrts
```

Results

Results for perf with mpirun on a single and 80 CPUs

```
ana-4.0.59-py3 [dubrovin@sdfmilan216:~/LCLS/con-py3]$ perf stat -e cache-references,cache-misses,cycles,
instructions,branches,branch-misses,faults,migrations,page-faults,L1-dcache-load-misses,L1-icache-load-misses,
dTLB-load-misses,iTLB-load-misses mpirun -n 1 python Detector/examples/test-scaling-mpi.py
...
```

Performance counter stats for 'mpirun -n 1 python Detector/examples/test-scaling-mpi.py':

```

4,448,830,552      cache-references:u                      (50.00%)
 90,374,312       cache-misses:u                      #    2.031 % of all cache refs (50.00%)
222,814,516,280    cycles:u                          (50.02%)
426,700,282,993    instructions:u                      #    1.92  insn per cycle      (50.01%)
58,876,394,584     branches:u                          (50.01%)
2,343,687,188      branch-misses:u                      #    3.98% of all branches     (50.01%)
 635,183          faults:u
 0               migrations:u
 635,183          page-faults:u
2,158,358,417      L1-dcache-load-misses:u              (50.00%)
 5,694,036         L1-icache-load-misses:u              (49.99%)
 4,282,821         dTLB-load-misses:u                   (49.99%)
 890,671          iTLB-load-misses:u                   (50.00%)
```

73.297275789 seconds time elapsed

69.795728000 seconds user

2.318007000 seconds sys

```
ana-4.0.59-py3 [dubrovin@sdfmilan216:~/LCLS/con-py3]$ perf stat -e cache-references,cache-misses,cycles,
instructions,branches,branch-misses,faults,migrations,page-faults,L1-dcache-load-misses,L1-icache-load-misses,
dTLB-load-misses,iTLB-load-misses mpirun -n 80 python Detector/examples/test-scaling-mpi.py
...
```

Performance counter stats for 'mpirun -n 80 python Detector/examples/test-scaling-mpi.py':

```

349,526,509,383    cache-references:u                      (50.01%)
 5,932,480,814     cache-misses:u                      #    1.697 % of all cache refs (50.00%)
18,768,444,974,036 cycles:u                          (50.00%)
33,983,153,714,284 instructions:u                      #    1.81  insn per cycle      (49.99%)
 4,684,730,635,234 branches:u                          (49.99%)
186,649,297,019    branch-misses:u                      #    3.98% of all branches     (50.00%)
 52,121,421        faults:u
 0               migrations:u
 52,121,421        page-faults:u
171,500,392,922     L1-dcache-load-misses:u              (50.00%)
267,672,856        L1-icache-load-misses:u              (50.00%)
339,145,247        dTLB-load-misses:u                   (50.01%)
 69,780,394        iTLB-load-misses:u                   (50.01%)
```

92.952500273 seconds time elapsed

6501.353593000 seconds user

410.844719000 seconds sys

Summary

number of mpi cores	cache- references	cache- misses	cycles	instructions	branches	branch- misses	faults	page- faults	L1- dcache- load- misses	L1- icache- load- misses	dTLB- load- misses	iTLB- load- misses	cmt
1	4,448,830,552	90,374,312	222,814,516,280	426,700,282,993	58,876,394,584	2,343,687,188	635,183	635,183	2,158,358,417	5,694,036	4,282,821	890,671	
80	349,526,509,383	5,932,480,814	18,768,444,974,036	33,983,153,714,284	4,684,730,635,234	186,649,297,019	52,121,421	52,121,421	171,500,392,922	267,672,856	339,145,247	69,780,394	
Ratio (80)/(1)	79.4	65.7	84.1	79.6	79.5	79.7	82.0	82.0	79.3	47.0	79.2	78.4	

2024-02-09 Test of milano216 host with command perf

Description

Use commands with changed list of counters like

```
perf stat -e stalled-cycles-backend,stalled-cycles-frontend,ls_l1_d_tlb_miss.all,l1_dtlb_misses,l1_data_cache_fills_all,bp_l1_tlb_miss_l2_tlb_miss.if2m,
bp_l1_tlb_miss_l2_tlb_miss,l2_dtlb_misses,l2_itlb_misses python test-scaling-subproc.py -8
```

Convert perf output to dict, present results in table.

Summary

(*) CPU numbers excludes weka FS.

number of CPU	stalled-cycles-backend	Ratio N/1	stalled-cycles-frontend	Ratio N/1	ls_l1_d_tlb_miss.all	Ratio N/1	l1_dtlb_misses	Ratio N/1	l1_data_cache_fills_all	Ratio N/1	bp_l1_tlb_miss_l2_tlb_miss.if2m	Ratio N/1	bp_l1_tlb_l2_tlb_miss
1	143,828614	1	230,987724	1	33,227437	1	32,845193	1	2179,469714	1	3,701	1	769,309
8	2105,881833	15	3421,108359	15	172,779030	5.2	173,508212	5.3	18216,564874	8.3	25,606	6.9	6124,897
16	8796,313234	61	8018,691890	35	327,892753	9.9	326,337183	9.9	34551,341060	15.8	55,331	14.8	12467,976
24*	10413,149941	72	10519,490870	46	491,673248	14.8	490,566093	14.9	51539,384297	23.6	78,433	21	17889,621
32	17251,055297	120	13858,554955	60	671,047247	20.2	666,230997	20.3	68736,842168	31.5	105,874	29	23936,978
56*	17892,504080	124	24120,493158	104	1136,778538	34.2	1135,448325	34.6	120696,775952	55.3	178,082	48	42679,843
64	27304,844238	190	27697,522017	120	1258,999729	37.9	1258,031354	38.3	141469,109046	64.9	201,330	54	50957,218
120*	45388,735746	316	46279,264661	200	2382,065820	71.6	2376,507106	73.3	264016,453328	121	375,699	102	93410,817

References

- [Scaling behavior of psana1 - Part 1 - det.calib method in multicore processing with mpi](#)
- [Scaling behavior of psana1 - Part 3 - test for reduced memory consumption](#)