

Tools/APS/example_rank_league

Example Intel Application Performance Snapshot: rank_league

- Prepare environment

```
module purge
```

```
# Load Intel compiler environment
```

```
source /software/all/toolkit/Intel_OneAPI/compiler/latest/env/vars.sh
```

```
# Load Intel MPI environment
```

```
source /software/all/toolkit/Intel_OneAPI/mpi/latest/env/vars.sh
```

- Build rank_league benchmark

```
mpicc -Ofast -xHost -ipo rank_league.c -o rank_league
```

- Jobscript jobscript.aps.sh

```
#!/usr/bin/bash
```

```
#SBATCH --partition=<...>
```

```
#SBATCH --nodes=4
```

```
#SBATCH --tasks-per-node=1
```

```
#SBATCH --time=10
```

```
# Prepare environment
```

```
module purge
```

```
# Load Intel compiler environment
```

```
source /software/all/toolkit/Intel_OneAPI/compiler/latest/env/vars.sh
```

```
# Load Intel MPI environment
```

```
source /software/all/toolkit/Intel_OneAPI/mpi/latest/env/vars.sh
```

```
# Load Application Performance Snapshot (APS) environment
```

```
source /software/all/toolkit/Intel_OneAPI/vtune/latest/apsvars.sh
```

```
# Set MPI Level of Detail
# See: https://www.intel.com/content/www/us/en/docs/vtune-profiler/user-guide-application.html
# To get information about transfers per communication. Set the APS_STAT_LEVEL value to 2
export MPS_STAT_LEVEL=2
```

```
# rank_league options
# test_type:  b - bandwidth
# output_type: s - statistics per rank - average, min, max
# loop_num:   number of loops per every round
RANK_LEAGUE_OPTIONS=( -t=b -o=s -l=20000 )
```

```
MPIRUN_OPTIONS=( -print-rank-map -binding domain=core )
```

```
mpirun "${MPIRUN_OPTIONS[@]}" aps ./rank_league "${RANK_LEAGUE_OPTIONS[@]}"
```

- Run benchmark rank_league with APS with batch system

```
sbatch rank_league.aps.job
```

- Job output

```
(hkn0004:0)
(hkn0005:1)
(hkn0006:2)
(hkn0007:3)
```

```
***** Running bandwidth test *****
Total number of rounds:      3
Total number of loops per round: 20000
Message size:                100000
*****
```

```
Round number      3
*****
```

RANK	MIN RESULT	RANK	MAX RESULT	RANK	AVERAGE
0	9880.39	2	16132.57	1	13536.02
1	9595.56	0	17246.61	3	12239.32
2	9697.62	0	17418.24	3	14390.22
3	9708.94	2	16937.74	1	12147.46

```
-----
Global statistics:
```

```
MIN      9595.56 between 1 and 0
MAX      17418.24 between 2 and 3
AVERAGE 13078.26
```

```
Intel(R) VTune(TM) Profiler 2023.1.0 collection completed successfully. Use the "aps --
```

- Generate APS report:

```
aps --report <...>/aps_result_20230526
```

```
Loading 100.00%
```

```
| Summary information
```

```
|-----
```

```
Application           : rank_league
Report creation date   : 2023-05-26 15:13:34
Number of ranks        : 4
Ranks per node         : 1
HW Platform            : Intel(R) Xeon(R) Processor code named Icelake
Frequency              : 2.39 GHz
Logical core count per node : 152
Collector type         : Driverless Perf per-process counting
Used statistics        : <...>/aps_result_20230526
```

```
|
```

```
| Your application might underutilize the available logical CPU cores
```

```
| because of insufficient parallel work, blocking on synchronization, or too much I/O.
```

```
|
```

```
Elapsed Time:                2.10 s
SP GFLOPS:                    0.00
DP GFLOPS:                    0.00
Average CPU Frequency:        3.38 GHz
IPC Rate:                     1.09
```

```
| Some of the individual values contributing to this average metric broke the
| issue threshold of the metric.
```

```
| Please use --counters or --metrics="Instructions Per Cycle Rate" reports for
| details.
```

```
MPI Time:                    1.79 s                87.40% of Elapsed Time
```

```
| Your application is MPI bound. This may be caused by high busy wait time
| inside the library (imbalance), non-optimal communication schema or MPI
| library settings. Explore the MPI Imbalance metric if it is available or use
| MPI profiling tools like Intel(R) Trace Analyzer and Collector to explore
| possible performance bottlenecks.
```

```
MPI Imbalance:                0.01 s                0.30% of Elapsed Time
```

```
Top 5 MPI functions (avg time):
```

```
MPI_Isend:                    0.66 s                32.39% of Elapsed Time
```

```
MPI_Irecv:                    0.62 s                30.40% of Elapsed Time
```

```
MPI_Init:                     0.34 s                16.68% of Elapsed Time
```

```
MPI_Barrier:                  0.12 s                 5.83% of Elapsed Time
```

```
MPI_Waitall:                   0.04 s                 2.11% of Elapsed Time
```

```
Physical Core Utilization:    0.95%
```

```
| The metric is below 80% threshold, which may signal a poor physical CPU cores
| utilization caused by: load imbalance, threading runtime overhead, contended
| synchronization, insufficient parallelism, incorrect affinity that utilizes
| logical cores instead of physical cores. Perform threading analysis with tools
| like Intel(R) VTune(TM) Profiler to discover why physical cores are
```

```

| underutilized.
|   Average Physical Core Utilization:          0.72 out of 76 Physical Cores
|   Memory Stalls:                             46.30% of Pipeline Slots
|   The metric value can indicate that a significant fraction of execution
|   pipeline slots could be stalled due to demand memory load and stores. See the
|   second level metrics to define if the application is cache- or DRAM-bound and
|   the NUMA efficiency. Use Intel(R) VTune(TM) Profiler Memory Access analysis to
|   review a detailed metric breakdown by memory hierarchy, memory bandwidth
|   information, and correlation by memory objects.
|   Cache Stalls:                             30.10% of Cycles
|   A significant proportion of cycles are spent on data fetches from cache. Use
|   Intel(R) VTune(TM) Profiler Memory Access analysis to see if accesses to L2 or
|   L3 cache are problematic and consider applying the same performance tuning as
|   you would for a cache-missing workload. This may include reducing the data
|   working set size, improving data access locality, blocking or partitioning the
|   working set to fit in the lower cache levels, or exploiting hardware
|   prefetchers.
|   DRAM Stalls:                             17.52% of Cycles
|   Some of the individual values contributing to this average metric broke the
|   issue threshold of the metric.
|   Please use --counters or --metrics="DRAM Stalls" reports for details.
|   Average DRAM Bandwidth:                   N/A
|   Data for this metric is not collected since it requires system-wide
|   performance monitoring. Make sure the sampling driver is properly installed on
|   your system: https://software.intel.com/en-us/vtune-amplifier-help-sep-driver.
|   Otherwise, enable a driverless Perf-based sampling collection by setting the
|   /proc/sys/kernel/perf_event_paranoid value to 0 or less.
|   NUMA:                                     0.12% of Remote Accesses
|   Vectorization:                           0.00%
|   Instruction Mix:
|     SP FLOPs:                               0.00% of uOps
|     DP FLOPs:                               0.00% of uOps
|     Non-FP:                                 100.00% of uOps
|     FP Arith/Mem Rd Instr. Ratio:           0.00
|     FP Arith/Mem Wr Instr. Ratio:           0.00
|   Disk I/O Bound:                           0.00 s
|   Memory Footprint:
|   Resident:
|     Per node:
|       Peak resident set size      :          123.00 MB (node hkn0005.localdomain)
|       Average resident set size   :          119.25 MB
|     Per rank:
|       Peak resident set size      :          123.00 MB (rank 1)
|       Average resident set size   :          119.25 MB
|   Virtual:
|     Per node:

```

```

Peak memory consumption      :      261.00 MB (node hkn0005.localdomain)
Average memory consumption   :      258.50 MB
Per rank:
Peak memory consumption      :      261.00 MB (rank 1)
Average memory consumption   :      258.50 MB

```

Graphical representation of this data is available in the HTML report: <...>/aps_report

- Generate APS Rank-to-rank communication matrix (requires MPS_STAT_LEVEL=4):

```

# in text format
aps --report -x <...>/aps_result_20230526
# or in html format
aps --report -x --format=html <...>/aps_result_20230526

```

Loading 100.00%

| Data Transfers per Rank-to-Rank Communication for all Ranks

Rank --> Rank	Time(sec)	Volume(MB)	Transfers
0000 --> 0001	0.11	4000.00	40008
0000 --> 0002	0.05	2000.00	20006
0000 --> 0003	0.05	2000.00	20006
0001 --> 0000	0.11	4000.00	40009
0001 --> 0002	0.05	2000.00	20006
0001 --> 0003	0.05	2000.00	20006
0002 --> 0000	0.07	2000.00	20007
0002 --> 0001	0.05	2000.00	20006
0002 --> 0003	0.11	4000.00	40008
0003 --> 0000	0.07	2000.00	20007
0003 --> 0001	0.05	2000.00	20006
0003 --> 0002	0.10	4000.00	40008
=====			
TOTAL	0.89	32000.00	320083
AVG	0.07	2666.67	26673