# Evaluation of benchmarks

June 23, 2020

## Throughput and strong-scaling benchmarks

All application benchmarks belong to one of two classes: *throughput* and *strong-scaling*.

### Throughput

- Use case: regular HPC research where a certain number of jobs need to be completed in order to finish a research project, without wasting core hours.
- Figure of merit (result): Number of jobs that run on the system in one day.
- Formula:

$$n_{\text{jobs}}(b) = \frac{t_{\text{day}}}{t(b)} \cdot \frac{N_{\text{tot}}}{N(b)} \tag{1}$$

  where:

  - $t_{\text{day}}$ = time in a day (86400 s)
  - $t$(b) = average time in seconds to run one job for benchmark $b$
  - $N_{\text{tot}}$ = total number of nodes tendered
  - $N(b)$ = number of nodes used to run benchmark $b$
  - $n_{\text{jobs}}(b)$ = number of jobs that can be run in a day for benchmark $b$. This result must be reported for each throughput benchmark in the benchmark matrix spreadsheet.
- All throughput benchmarks have a **minimum performance** to ensure that more than one node needs to be used.

The throughput tests represent a trade-off between maximizing the number of nodes in the system, minimizing the time it takes to run one job and minimize the number of nodes used for each job. Each throughput benchmark case has a specified minimum performance which most likely requires it to be run on more than one node.

### Strong Scaling

- Use case: Researcher needs to finish a massively parallel job in the shortest possible time. The researcher has access to the entire system.
- Figure of merit (result): Number of jobs (each job solves the problem in the shortest possible time) that run on the system in one day.
- Formula:

$$n_{\text{jobs}}(b) = \frac{t_{\text{day}}}{t(b)} \tag{2}$$

  where:

  - $t_{\text{day}}$ = time in a day (86400 s)
  - $t(b)$ = average time in seconds to run one job in the shortest possible time for benchmark $b$
  - $n_{\text{jobs}}(b)$ = number of jobs that can be run in one day for benchmark $b$. This result must be reported for each strong-scaling benchmark in the benchmark matrix spreadsheet.

In the strong scaling benchmarks the goal is simply to minimize the runtime of the benchmark cases by using as many nodes/cores/devices as required to maximize performance.

The calculation of the throughput and strong-scaling figures of merit is clarified in the documentation for each benchmark case.

# Scoring

The result, i.e. figure of merit (FOM), obtained from each benchmark case will be used to distribute points corresponding to historical usage on previous PDC/SNIC systems called Maximum Points, and the final benchmark score is a sum of these points.

Points for each benchmark case will be calculated relative to the best result reported by any tenderer, with the best result awarded the Maximum Points for that case. For instance, if tenderer A obtains the highest figure of merit (largest number of jobs per day) for the Gromacs case "EAG1 membrane protein" on Phase 1, tenderer A will obtain 3.5 points for that benchmark case (the Maximum Points). If tenderer B obtains a figure of merit which is 75% of that reported by tenderer A, tenderer B will obtain 0.75*3.5 = 2.62 points for that case. To avoid a situation where non-competitive tenders influence the ranking of the best tenders, an iterative elimination procedure will be applied as further described below.

### Weighting by operational time for Phase 2

All benchmarks results for Phase 2 will be weighted with how long Phase 2 is planned to be operational at PDC. (For Phase 1 the installation window is so small that this type of calculation is not meaningful). So the actual figure of merit of the benchmark case (number of jobs per day) will be multiplied with time from the guaranteed date for the acceptance until 2026-08-31 in days. The relative points for each benchmark case will be computed based on this weighted figure of merit (see below how the relative points will be computed). Thus, a system which will be operational for a longer duration will get a proportionally higher points than a system of the same capacity that is installed later. For example, a date for approved acceptance of 2022-03-15 would give an installation time for Phase 2 of 1630 days and a date for approved acceptance of 2021-10-15 would give an installation time of 1781 days.

The weighting with operational time is applied before the relative points are calculated. For instance, if tenderer A obtains 100 jobs/day for the CP2K-Quickstep benchmark and the number of days from the guaranteed date for acceptance until 2026-08-31 is 1500, the total number of jobs will be 100*1500 = 150000. If tenderer B obtains a better performance of 110 jobs/day but the total number of days is 1200, the total number of jobs will be 110*1200 = 132000. In this case, if tenderer A obtains the largest number of jobs out of all tenderers, tenderer A will receive 9.0 points (the Maximum Points) and tenderer B will obtain 132000/150000*9.0 = 7.92 points.

### Available points

Tables 1 and 2 summarize the Maximum Points from each benchmark application and case for Phase 1 and Phase 2, respectively.

Table 1: Maximum points available from benchmark cases in Phase 1

| Application | Case | Max points |
|---|---|---|
| Gromacs | EAG1 membrane protein | 3.5 |
| | Membrane protein Aquaporin | 3.5 |
| Nek5000 | Pipe simulation case | 7.0 |
| VASP | GaAsBi 512 atoms | 5.0 |
| PowerFLOW | BC_01 | 5.0 |
| SingleFFT | FFT synthetic benchmark | 1.0 |

## Ranking process

An iterative elimination process will be applied to determine the winning tender. This will eliminate the risk of irrelevant non-competitive tenders affecting the ranking of competitive tenders, which could

Table 2: Maximum points available from benchmark cases in Phase 2

| **Application** | **Case** | **Max points** |
|---|---|---|
| Gromacs | EAG1 membrane protein | 4.5 |
| | Membrane protein Aquaporin | 4.5 |
| PyFR | NACA0021 Aerofoil single-precision | 4.0 |
| | NACA0021 Aerofoil double-precision | 4.0 |
| CP2K-Quickstep | Linear scaling DFT | 9.0 |
| SingleFFT | FFT synthetic benchmark | 4.0 |

happen if only one round of relative points are evaluated and irrelevant tenders with low total score happen to obtain Maximum Points on certain benchmark cases (and thus affect the point difference between relevant tenders).

The score obtained from the other evaluation criteria in section XXX will be added to the relative benchmark points before the elimination is performed.

The procedure will be as follows:

1. Relative points are calculated for all tenders for all benchmark cases, and a weighted sum is computed to obtain the total benchmark score for each tender.
2. The total benchmark score is added to the score from the other evaluation criteria to give a Total Score.
3. Tenders are ranked according to their Total Score, and the tender with the lowest Total Score is eliminated.
4. Repeat from step 1 until only one tender remains.

**Example**

The following table illustrates a hypothetical scenario with 3 tenders and 3 benchmarks: bench1, bench2 and bench3. The figure of merit (FOM) for each benchmark has the unit jobs/day, but the absolute numbers do not matter since relative points will be calculated in the next step.

Table 3: Figure of merit (FOM) for 3 tenderers and 3 benchmarks

| Tender | FOM bench1 | FOM bench2 | FOM bench3 |
|---|---|---|---|
| A | 140 | 80 | 500 |
| B | 130 | 110 | 450 |
| C | 30 | 190 | 200 |

Relative points are now calculated according to the winner in each benchmark, see table 4. In this scenario the 3 benchmarks have Maximum Points 5.0, 7.5, 10.0, respectively. Tender C achieves the best FOM and therefore maximum points for benchmark 2, but obtains low FOMs for the other two benchmarks and thus low total benchmark score.

Table 4: Relative points based on the benchmark results in table 3

| Tender | Points bench1 | Points bench2 | Points bench3 | Total benchmark score |
|---|---|---|---|---|
| A | 5.0 | 3.158 | 10.0 | 18.158 |
| B | 4.643 | 4.342 | 9.0 | 17.985 |
| C | 1.071 | 7.5 | 4.0 | 12.571 |

The Total Score is now obtained by summing the total benchmark points with the points from the should-requirements. Let's say tenders A, B and C obtain 10.0, 10.0 and 6.0 points, respectively, from the should-requirements. Their Total Scores are then:

- Tender A: 18.158 + 10.0 = 28.158

- Tender B: 17.985 + 10.0 = 27.985
- Tender C: 12.571 + 6.0 = 18.571

Tender A comes out on top in this round, but an elimination step is now performed. Tender C has the lowest Total Score and is thus eliminated. This leaves tenders A and B.

The benchmark performance remains the same as in table 3, but new relative points are now calculated, see table 5.

Table 5: New relative points and Total benchmark score based on the benchmark results in table 3 after tenderer C has been eliminated.

| Tender | Points bench1 | Points bench2 | Points bench3 | Total benchmark score |
|---|---|---|---|---|
| A | 5.0 | 5.455 | 10.0 | 20.455 |
| B | 4.643 | 7.5 | 9.0 | 21.143 |

Their Total Score after adding the should-requirement points are:

- Tender A: 20.455 + 10.0 = 30.455
- Tender B: 21.143 + 10.0 = 31.143

If more tenders had been present this process would have been repeated until only two tenders remained. In this case, tender B receives the highest Total Score and thus wins the bid.

In this hypothetical scenario the order between tenders A and B changed after tender C was eliminated. In reality this might be a relatively rare occurrence. In a simulated bidding process with 5 tenders we observe such changes to occur around 10% of the time with random benchmark results for 7 benchmarks.