Proposal title

PI   CoPI
affiliation,   affiliation

Abstract

This is the abstract of the proposal ...

1   Background and Significance

The project proposal should be no longer than *10 A4 pages* including graphs and references, and must contain the following information:

- Abstract

- Background and significance

- Scientific goals and objectives

- Description of the research methods, algorithms, and code

- Parallelization approach, memory requirements

- Representative benchmarks and scaling

- Performance analysis (link: see how to do it)

- Resource justification (annual node hours and disk space)

- Project plans: tasks and milestones

- Visualization, pre- and post-processing needs

- Previous results are mandatory! Please list allocations requested, granted and used in your previous project (if applicable)

- Research publications that resulted from these allocations

- Development and debugging requirements (not mandatory)

Therefore you should follow the structure used in this template, which reflects the official requirements. Please check the page with the general instructions for Proposal Submission on the CSCS Website for more details.

2   Scientific Goals and Objectives

...

3   Research Method, Algorithms and Code

Please insert in this section a comprehensive explanation with a brief list of libraries used: fftw, parallel-hdf5, petsc, ...

4 Parallelisation Approach and Memory Requirements

Please report in this section the parallel approach employed to address the proposed computational study with the selected code. For instance report if using MPI distributed parallelism or hybrid MPI/OpenMP, including details like the type of MPI communication, halo cell exchange and which part of the code is taking advantage of shared memory parallelism. You should also point out which portions of the selected code used GPU or OpenACC/CUDA ...

5 Representative Benchmarks and Scaling

Please report in this section the mandatory strong scaling tests performed using the selected code to address the proposed study with a representative system to be investigated during the research work. The scope of this test is to choose the most efficient job size to run the performance analysis on your representative system that will be reported in the next section. You should therefore select meaningful job sizes for your representative system, compatible with reasonably short runtimes: the lowest number of nodes might be determined by memory and wall-time constraints, while the highest number of nodes tested should allow you identify the job size at which you reach $\sim 50\%$ of the parallel efficiency, given by the ideal scaling. Weak scaling tests might be provided as well, in addition to the srong scaling scalability data.

Figure 1 shows the scalability plot for our example system: we start the scalability study with a job size of 8 nodes, whose runtime will be used as reference to compute the speed-up of larger job sizes. We then proceed doubling the number of nodes and checking the corresponding speed-up, until we are sure to have reached the 50% limit in parallel efficiency.
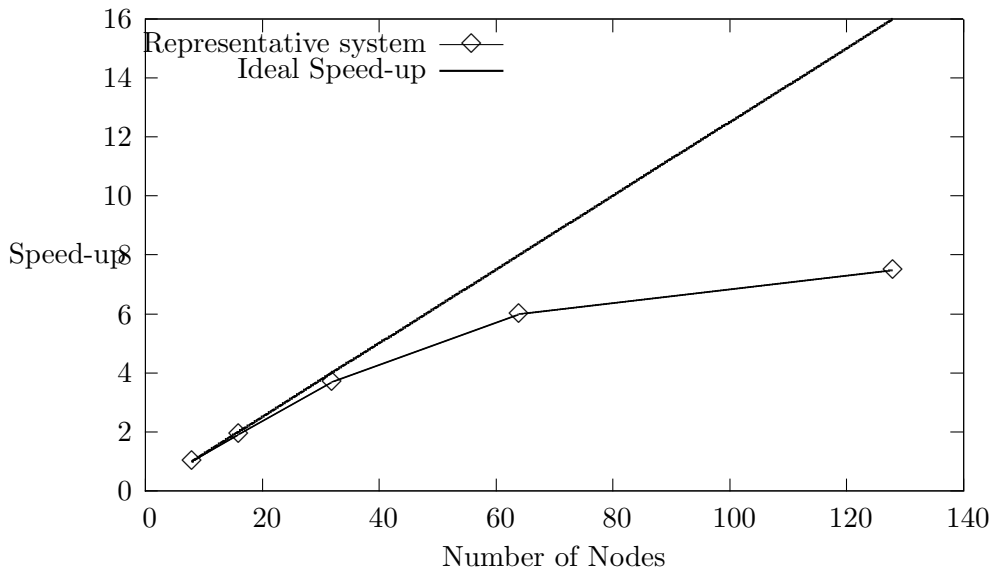

Figure 1: Scalability test: strong scaling efficiency vs. ideal speed-up for the representative system

6 Performance Analysis

We report in the present section a summary of the performance analysis conducted on the representative system at the job size selected in the previous section, which reached $\sim 50\%$ parallel efficiency. We have followed the instructions available on the section Performance Report of the CSCS User Portal, in order to instrument the executable of the selected code with *Cray Performance and Analysis Tools*.

The results of the run with the instrumented executable are the report file with extension `.rpt` which we attach at the time of the proposal submission and the larger apprentice file with extension `.ap2`: we will save both files on our `$HOME` or `$PROJECT` folder and make them readable to allow CSCS technical reviewers inspect them. We have documented a single production system: in case your project is split into multiple production runs with the same resources requirements, then please choose the most representative of them. Please find below the required summary data extracted from the report file:

```
Experiment:                   lite  lite/gpu
Number of PEs (MPI ranks):     16
Numbers of PEs per Node:        1  PE on each of  16  Nodes
Numbers of Threads per PE:   1,114
Number of Cores per Socket:    12
Execution start time:  Tue Mar 28 15:15:55 2017
System name and speed:  nid02294  2601 MHz (approx)
Intel haswell CPU  Family:  6  Model: 63  Stepping:  2


Avg Process Time:      2,100 secs
High Memory:        13,977.3 MBytes     873.6 MBytes per PE
I/O Read Rate:     67.110363 MBytes/sec

| 31.8% |   664.415775 |            -- |   -- |     35,648.0 |MPI_SYNC
|  2.8% |    58.511390 |            -- |   -- | 14,458,788.1 |MPI

| 59.2% | 1,236.266484 | 110.728787 |  8.8% |          1.0 |USER
```

7   Resources Justification

The computation of the required time should start with the node hours required for a single timestep. As an example, the values in Table 1 sum up to 107500 node hours.

	simulation type 1	simulation type 2
number of simulations	2	3
node hours per timestep	1	1.5
number of timesteps per simulation	20000	15000
node hours	40000	67500

Table 1: Derivation of node hours

8   Project Plans: Tasks and Milestones

...

9   Visualization, pre- and post-processing

...

10   Results from Previous Allocations

...

11 Research publications from Previous Allocations

...

12 Development and debugging requirements

...

References