

# This is the title of the Proposal

Principal Investigator<sup>1</sup>, First Collaborator<sup>2</sup>, and Last Collaborator<sup>1,2</sup>

<sup>1</sup>Affiliation of the PI

<sup>2</sup>Other affiliation

## Abstract

This is the abstract of the proposal: in this document we provide a template for CSCS Production Project Submission with guidelines, focusing in particular on sections **Representative benchmarks and Scaling**, **Performance Analysis** and **Resource Justification**.

## 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
- Resource justification (annual node-hours and disk space)
  - ★ Visualization, pre- and post-processing needs
  - ★ Development and debugging requirements
- Project plan: tasks and milestones
- Previous results

Please follow the structure used in this template, which reflects the requirements reported on [Production Projects Submission](#).

## Scientific Goals and Objectives

...

## Research Methods, Algorithms and Code

Please insert in this section a description of the methods and algorithms of the code adopted for your computational study, with a brief list of the main scientific libraries employed.

## Parallelization Approach and Memory Requirements

Please present in this section the parallel approach adopted to address the proposed computational study with the selected code: in general community codes publish this information on their web sites. For instance, [CP2K home page](#) reports that *CP2K is written in Fortran 2003 and can be run efficiently in parallel using a combination of multi-threading, MPI, and CUDA*. If you don't use a community code, please report if the code employs MPI distributed parallelism or hybrid MPI/OpenMP, which type of MPI communication has been implemented and if it makes use of shared memory parallelism, GPU accelerators or OpenACC/CUDA.

## Representative Benchmarks and Scaling

Please report in this section the results of the mandatory strong scaling tests performed with the selected code: please report scaling data and plot for every representative system of your project. The goal is to choose the most efficient job sizes to run the performance analysis.

You should select meaningful job sizes to simulate the representative systems, compatible with reasonably short runtimes: the lowest number of nodes is determined in general by memory and wall time constraints, while the highest node counts should let you identify the job size at which you reach  $\sim 50\%$  of the parallel efficiency with respect to ideal scaling. Weak scaling tests might be provided as well, in addition to the required strong scaling data.

Table 1 reports the wall time in seconds and the corresponding speed-up for a single representative system. Figure 1 shows the scaling plot: we started the scaling test on 2 nodes, taking this runtime as a 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  $\sim 50\%$  limit in parallel efficiency (16 nodes in the small example below).

Nodes	Wall time (s)	Speed-up
2	1022	1.00
4	476	2.15
8	339	3.01
16	210	4.87
32	206	4.96

Table 1: Wall time and speed-up

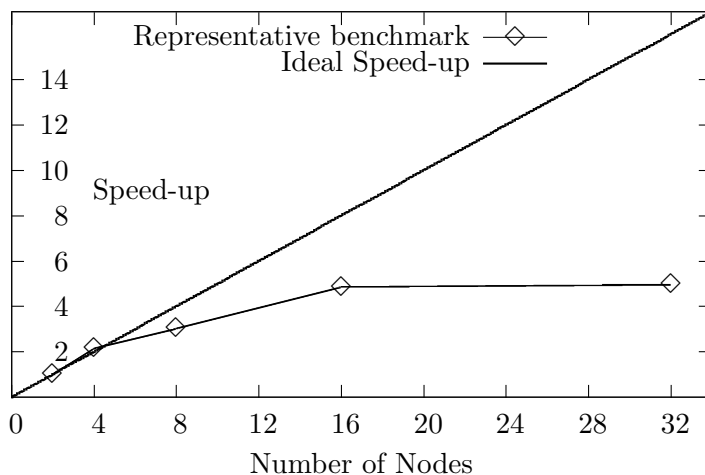


Figure 1: Strong scaling vs. ideal speed-up

## Performance Analysis

Please report here a summary of the performance analysis conducted on each representative system at the optimal job sizes selected in the previous section, which reached  $\sim 50\%$  parallel efficiency. You should have already instrumented the executable with *Cray Performance and Analysis Tools*, following the instructions of [Performance Report](#).

The performance analysis job will create a report text file with extension `.rpt` and a larger apprentice binary file with extension `.ap2`. Please make these two files available for inspection, either by enclosing them at submission time or indicating where they can be accessed for reading under `$HOME` or `$PROJECT` (not `$SCRATCH`).

The summary data can be extracted using the following commands on the report text file:

```
grep -A 7 CrayPat/X <report>.rpt
grep \|USER <report>.rpt
grep \|MPI <report>.rpt
grep \|Total <report>.rpt
```

The summary should look like the example below:

```
CrayPat/X:  Version 6.4.5 Revision 87dd5b8  01/23/17 15:37:24
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)

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

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

100.0% | 2,086.808412 |          -- |    -- |        18,723,148.8 |Total
100.0% |  1.89 | 105,946 |   287.62 | 75,246 |Total
56.092035 | 3,764.356845 |  67.110363 | 62,097,047.0 |    63.57 |Total
0.151159 |  2.949494 | 19.512511 | 74,334.0 |    41.61 |Total
```

The first command extracts general information on the job, then we extract the statistics of USER and MPI functions; the last command reports the Total of each Table (functions, accelerator, read and write statistics).

## Resource Justification

The request of the annual amount of node-hours should be clearly linked with the node-hours used by the representative benchmarks: the number of node-hours consumed by a simulation is computed multiplying the number of nodes by the wall time expressed in hours. CrayPAT adds an overhead to the wall time, therefore you cannot use that timing to justify your request.

In the small example used throughout this template, the optimal job size of the representative benchmark is 16 nodes and the corresponding wall time reported in Table 1 is 210 s, which correspond to  $\sim 0.933$  node-hours, as a result of the multiplication  $16 \text{ nodes} \times \frac{210s}{3600 \frac{s}{hour}}$ . The benchmark is short and represents in general a small number of iterations (cycles, timesteps or an equivalent measure), while in a real production simulation we will need to extend it.

Therefore we will estimate how many iterations should be necessary to complete a simulation in production. Furthermore, the project plan might contain multiple tasks, each of them requiring several sets of simulations to complete: the annual resource request will sum up the corresponding node-hours obtained multiplying all the factors reported in Table 2.

	First task	Second task
Simulations per task	2	4
Iterations per simulation	5000	10000
node-hours per iteration	0.933	0.933
Total node-hours	9333	37333

Table 2: Justification of the resource request

The resource request of this small example will sum up to a total of 46666 annual node-hours, summing the node-hours estimated to complete the first and the second task of the project (Table 2), in agreement with the **Project Plan**.

You should present in this section your request for long term storage as well, explaining your needs based on the I/O pattern of the representative benchmarks reported by the performance analysis.

### Visualization, pre- and post-processing

Please insert in this subsection the optional requirements for visualization, pre- and post-processing.

### Development and debugging

Please insert in this subsection the optional requirements for development and debugging.

## Project Plan: Tasks and Milestones

Please report tasks and milestones of your project. When describing your project development, aside from laying out the tasks that take you from beginning to end, please mark key dates as well. An easy way to do this graphically is through the use of a Gantt chart: milestones charts are also useful to determine more accurately whether or not a project is on schedule.

## Results from Previous Allocations

Please list here your past requests, granted projects and used allocations (if applicable). You should also include a list of research publications that resulted from past allocations.

## References

...