B. Node performance analysis

algorithmically optimal implicit finite element solvers

granularity of the largest problem.

scalability, an impressive number considering the coarse

on 96 racks with 1,572,864 cores (granularity of 5K

Starting from one rack with 16,384 cores (granularity

would be required.

advantages of adaptive implicit solvers: adaptivity results

negligible time for I/O and problem setup stem from the

negligibility of I/O and setup time. The I/O for writing

output data has to be performed only once at the end of

scalability runs would produce

output.

weak scalability for the linear solver only

ideal speedup (efficiency baseline is the 1 rack result). We report both

iterations. Numbers along the graph lines indicate efficiency w.r.t.

96 racks. Performance is normalized by time and number of GMRES

Figure 5:

Billions DOF / s per GMRES it.

16,384    32,768    65,536   131,072   262,144   524,288 1,048,576

10

10

10

Contrary to conventional wisdom, this shows that

In Figure 6, we show strong scalability results for

Prerequisites for the course are some (serial) programming experience with C,

C++ (we will mostly use C and some basic C++ in class) or FORTRAN, some familiarity

with numerical methods and knowledge of basic UNIX commands. In case of doubt, please

come by or send an email to the instructors.

Required work: This will be a hands-on class, with several parallel (and serial) computing

assignments. Your active participation is crucial since you will have to explore material by

yourself and try things out. There will be individual larger final projects at the end, which

you can tackle by yourself or with a partner. Students who have code they want to parallelize

or speed up can use that for their final project.

Intended topics: Algorithms (matrix-matrix and matrix-vector multiplication, sorting, tree al-

gorithms, Jacobi smoothing and multigrid); Computer architectures (CPUs, accelerators,

memory hierarchy, parallel programming models, networks); Programming (single-core per-

formance optimization, OpenMP, MPI, Cuda); Tools (make, git, valgrind, shell scripting,

visualization).

Literature and Organization: We will regularly post links to online reading material and re-

sources which you are required to study. The class will be organized using Slack. If you are

signed up for the class you will get an invitation for the Slack group, if you are auditing let

us know and we will add you.