# 1 Overview

For spatial discretization implicit schemes are used. Hence, the calculation of a derivative always involves communication amongst all processors in a line along which a derivative is computed. From ad-hoc considerations it is not clear which is the optimum two-dimensional domain-decomposition. While for small numbers of cores one would expect the network latency of an MPI-Call to dominate, for larger numbers it is the number of processors involved in the call that makes MPI-calls expensive. Therefore, below a certain threshold, a 1D decomposition is expected to be beneficial. Where this turnover takes place is subject to many factors such as the CPU clock speed, network latency, MPI-implementation, number of grid points, etc. This section summarizes scaling tests that have been carried out on various machines, in particular the clusters `jugene@fz-juelich.de` and `blizzard@dkrz.de` in Aachen, respectively Hamburg.

Performance of the DNS code has been measured for various geometries varying the total number of grid points by two orders of magnitude. We define $Q := q_x \times q_y \times q_z$, the number of grid points, as a measure of the size of the simulations and choose in the following to label simulations by $Q$. To label the simulations, we use the common abbreviations

$$\mathrm{k} = 2^{10}; \qquad \mathrm{M} = 2^{20}; \qquad \mathrm{G} = 2^{30} \tag{1}$$

for readability. The memory necessary to save one 3D array in double precision data format is $8 \times Q$ Byte.

The scaling of the code is discussed in terms of Speedup $S$ and Efficiency $\eta$ defined as

$$S := \frac{T}{T_{\mathrm{ref}}} \qquad \eta = \frac{T}{T_{\mathrm{ref}}} \times 100\%. \tag{2}$$

# 2 Scaling on the cluster `jugene@fz-juelich.de`

## 2.1 Overview

Many-core scaling properties (up to 32k) of the DNS code, Version 5.6.6 on the machine `jugenefz-juelich.de` were investigated. Linear scaling is observed for up to 4096 MPI-Tasks using about $2 - 4 \times 2^{20}$ grid points per processor. The maximum efficiency is usually reached when simulations are carried out within one mid-plane. In this case for the standard domain sizes of up to 4Gpoints, a slightly super-linear scaling is observed. The code scales linearly up to 4096 processors for large domains of 24G-32G.

## 2.2 Setup

For measurements on `jugene`, the code has been instrumented to measure the real time that is necessary to perform one sub-step of the five-step Runge-Kutta time-stepping scheme. This corresponds essentially to the evaluation of the right-hand-side term of the governing equations solved. The instrumented code has been run for 2 iterations, i.e. 10 Runge-Kutta 'sub'-steps and variance the variance of measured times was found to be of the order of 1%. All simulations have been run in SMP mode using 4OpenMP threads. The cases considerer here are listed in Table 1.

## 2.3 Strong Scaling

The total number of cores $N$ available to a simulation are distributed as $N =$`ims_npro` $=$ `ims_npro_k` $\times$ `ims_npro_i` in the directions of $i$ ($x$) and $k$ ($z$). A series of measurements has been carried out to determine the optimum configuration for the six cases `1024x0192` - `4096x2048` listed in table 1. Scaling matrices are listed in Figure 1. In these matrices the number of processors is constant along diagonals from the lower left to the upper right.

| Name | $N_x$ | $N_y$ | $N_z$ | $Q$ |
|---|---|---|---|---|
| `1024x0384` | 1024 | 1024 | 384 | 384M |
| `2048x0192` | 2048 | 2048 | 192 | 768M |
| `2048x1024` | 2048 | 2048 | 1024 | 4G |
| `3072x1536` | 3072 | 3072 | 1536 | 12G |
| `4096x1536` | 4096 | 4096 | 1536 | 24G |
| `4096x2048` | 4096 | 4096 | 2048 | 32G |

Table 1: Geometry and labels of the 3D cases.

In every matrix, the diagonal for $N = 1k$ is outlined by solid borders. In each of these diagonals, the best and worst configurations are marked by green, respectively red, color. In these matrices, the speed-up and efficiency is for strong scaling, and always calculated with respect to the runtime for the lowest number of cores with the lowest `ims_npro_i`. Efficiency and speed-up are calculated as above.

### 2.4 Scaling from 32 to 8192 nodes

A strong scaling analysis over the entire range of nodes from 32 to 8192 is not possible. Hence, we restrict ourselves to a scaling analysis where we consider the number of grid points that is handled per processor and time unit. A straightforward definition for a metric of Performance $P$ is

$$P := \frac{Q}{NT}. \tag{3}$$

$P$ is a metric that makes performance comparable over *almost* arbitrary problem sizes and numbers of cores. Note, that the caveat here is the operation count for the Fourier transforms which goes as $q_i \log q_i$ and $q_i \approx Q^{1/3}$ if domains are expanded by the same factor in each direction. Hence, for the overall operation count of the Fourier transforms $\Sigma_{\mathrm{FFT}}$ we get

$$\frac{\Sigma_{\mathrm{FFT}}}{q_i^2} \propto 3 q_i \log q_i = Q^{1/3} \log Q \Rightarrow \Sigma_{\mathrm{FFT}} \propto Q \log Q. \tag{4}$$

For the range of $Q$ considered here, this effect is, however, small since the super-linear contribution of $\Sigma_{\mathrm{FFT}}$ is only $\frac{\log 32\mathrm{G}}{\log 384\mathrm{M}} \approx 1.2$ and the FFT accounts for a negligible part of the computational time ($1\% - 5\%$ of the computational part, which is 0.5%-2.5% of the overall time). The operation count of the

Given $P$, one can compute a virtual efficiency and speed-up $\eta_v$ and $S_v$ as

$$\eta_v := \frac{P}{P_{\mathrm{ref}}}; \qquad S_v := \frac{N}{N_{\mathrm{ref}}} \frac{P}{P_{\mathrm{ref}}} \tag{5}$$

with respect to a reference. Here, we use the same reference for all simulations and cases namely the `32 x 1` decomposition of the case with $Q = 384\mathrm{M}$. For larger numbers of cores $N$ and simulation sizes $Q$ we always choose the optimum configuration which is marked by green color in Figure 1.

The scaling as described in the above paragraph is summarized in Table 2 where the real-time for one Runge-Kutta substep in hundredth of a second as well as $P$, $\eta_v$ and $S_v$ for each of the configuration that have been measured is listed. Figure 3 shows the speed-up $S_v$. In the upper panel there is one line for each cases. Note, that the definition of $S_v$ does **not imply that cases start on the linear scaling line**. In this case, it is part of the measurements and simply means, that $\eta_v \approx 100\%$ or $P \approx P_{\mathrm{ref}}$.

## 3 Scaling on the cluster `blizzard@dkrz.de`

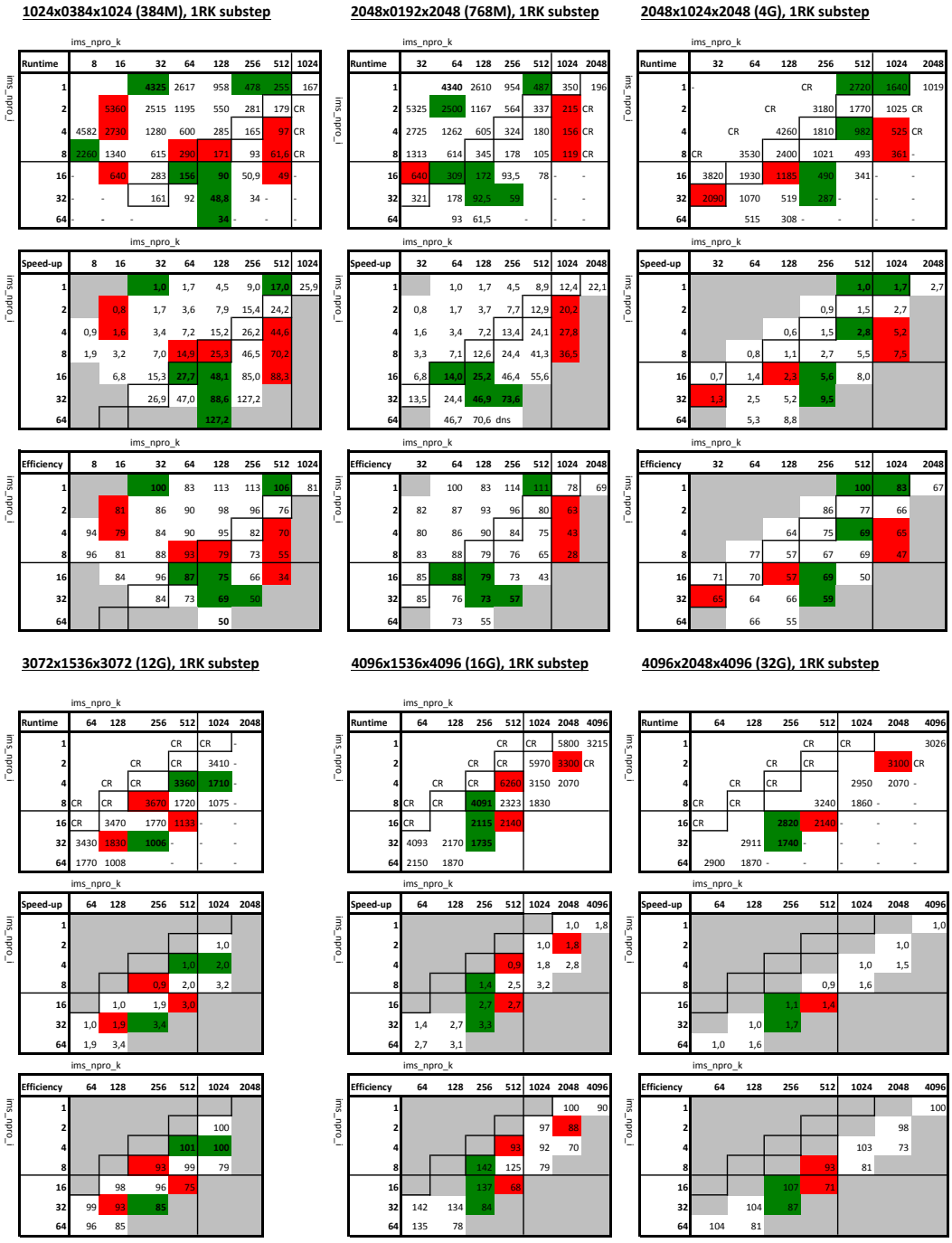


Figure 1: Matrices for scaling and 2D domain decomposition. Time is in hundredth of a second per single RK-substep. In the time-matrices, simulations that crashed because of too little memory (above diagonals) and page problems (below diagonals) are marked by CR. If no measurement for a certain configuration was attempted, the cell is left empty. For speed-up and efficiency all configurations not available are marked gray.

**T**

| N [k] \ G [M] | 384 | 768 | 4.096 | 13.824 | 24.576 | 32.768 |
|---|---|---|---|---|---|---|
| 1/32 | 4.325 | | | | | |
| 1/16 | 2260 | 4340 | | | | |
| 1/8 | 1340 | 2500 | | | | |
| 1/4 | 478 | 954 | | | | |
| 1/2 | 270 | 487 | 2720 | | | |
| 1/1 | 156 | 309 | 1640 | | | |
| 2/1 | 90 | 172 | 982 | 3410 | | |
| 4/1 | 48,8 | 93 | 490 | 1710 | 2115 | 2820 |
| 8/1 | 34 | 59 | 287 | 1006 | 1400 | 1740 |

**Efficiency w.r.t 32 cores of smallest case**

| N [k] \ G [M] | 384 | 768 | 4.096 | 13.824 | 24.576 | 32.768 |
|---|---|---|---|---|---|---|
| 1/32 | 100% | | | | | |
| 1/16 | 96% | 100% | | | | |
| 1/8 | 81% | 87% | | | | |
| 1/4 | 113% | 113% | | | | |
| 1/2 | 100% | 111% | 106% | | | |
| 1/1 | 87% | 87% | 88% | | | |
| 2/1 | 75% | 79% | 73% | 71% | | |
| 4/1 | 69% | 73% | 74% | 71% | 102% | 102% |
| 8/1 | 50% | 57% | 63% | 60% | 77% | 83% |

**S (Speed-up)**

| N [k] \ G [M] | 384 | 768 | 4.096 | | 24.576 | 32.768 |
|---|---|---|---|---|---|---|
| 1/32 | 1,0 | n/a | | | | |
| 1/16 | 1,9 | 2,0 | | | | |
| 1/8 | 3,2 | 3,5 | | | | |
| 1/4 | 9,0 | 9,1 | | | | |
| 1/2 | 16,0 | 17,8 | 17,0 | | | |
| 1/1 | 27,7 | 28,0 | 28,1 | | | |
| 2/1 | 48,1 | 50,3 | 47,0 | 45,7 | | |
| 4/1 | 88,6 | 93,0 | 94,1 | 91,1 | 130,9 | 130,9 |
| 8/1 | 127,2 | 146,6 | 160,7 | 154,8 | 197,7 | 212,1 |

**Q/T/N Megapoints per k-procs per second**

| N [k] \ G [M] | 384 | 768 | 4.096 | 13.824 | 24.576 | 32.768 |
|---|---|---|---|---|---|---|
| 1/32 | 2,84 | n/a | | | | |
| 1/16 | 2,72 | 2,83 | | | | |
| 1/8 | 2,29 | 2,46 | | | | |
| 1/4 | 3,21 | 3,22 | | | | |
| 1/2 | 2,84 | 3,15 | 3,01 | | | |
| 1/1 | 2,46 | 2,49 | 2,50 | | | |
| 2/1 | 2,13 | 2,23 | 2,09 | 2,03 | | |
| 4/1 | 1,97 | 2,06 | 2,09 | 2,02 | 2,90 | 2,90 |
| 8/1 | 1,41 | 1,63 | 1,78 | 1,72 | 2,19 | 2,35 |

Figure 2: Scaling results in term of Speed-Up $S_W$, Efficiency $E_W$ and Performance $P$ as defined above.

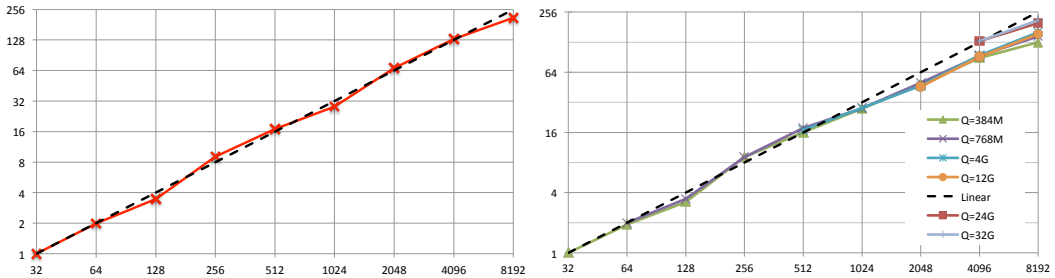


Figure 3: Upper panel: $S_W$ versus number of nodes for all geometries. Lower panel: maximum speedup for a given number of processors $\max(S_W)|_N$; axes as in upper panel.