Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Bachelorthesis

Title

vorgelegt von

TJARK SIEVERS

Fakultät: Mathematik, Informatik und Naturwissenschaften
Fachbereich: Physik
Studiengang: Physik
Matrikelnummer: 7147558
Erstgutachter:
Zweitgutachter:

Kurzzusammenfassung

Danksagung

Motivation

Inhaltsverzeichnis

Kurzzusammenfassung **i**

Danksagung **iii**

Motivation **v**

I Physical and Computational Basics **1**
- I.1 The electronic structure problem 1
- I.2 Density Functional Theory 1
  - I.2.1 Hohenberg-Kohn theorems 1
  - I.2.2 Kohn-Sham equations 2
- I.3 Parallel computing 2
  - I.3.1 On scalabilty 2
- I.4 QUANTUM ESPRESSO 3

II Results **5**
- II.1 Parallelisation of self-consistent calculations of electronic-structure properties . 5
  - II.1.1 First scaling tests 5
  - II.1.2 Different compilers 6
  - II.1.3 Using the parallelisation parameters of QUANTUM ESPRESSO 6
  - II.1.4 Comparison with calculations on the HLRN cluster 6
  - II.1.5 Scaling over 40 processors 6
  - II.1.6 Conclusion: Parameters for optimal scaling 6

Literaturverzeichnis **7**

Conventions

Throughout the text of this thesis scalars are written in italic s, vectors in bold italic $\boldsymbol{v}$ and matrices in bold $\mathbf{M}$ fonts. The summation/multiplication over nearest neighbour sites i and j is $\langle ij\rangle$ as a subscript to $\sum/\prod$. Furthermore, Hartree atomic units are used in general and only in selected instances it is deviated from this: $\hbar = m_\mathrm{e} = \mathrm{e} = {}^{4\pi}/_{\varepsilon_0} = 1$.

I Physical and Computational Basics

I.1 The electronic structure problem

In solid state physics, one general problem we are concerned with is finding the properties of the ground state of an isolated system of N interacting electrons in an external potential. The system is described by the Schrödinger equation

$$\hat{H}\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) = E\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) \tag{I.1}$$

with the Hamiltonian

$$\begin{aligned} \hat{H} &= \hat{T}_e + \hat{V}_{n-e} + \hat{V}_{e-e} + \hat{V}_{n-n} \qquad \text{(I.2)} \\ &= -\sum_i \frac{1}{2}\nabla_i^2 - \sum_i \sum_\alpha \frac{Z_\alpha}{|\mathbf{r}_i - \mathbf{R}_\alpha|} + \frac{1}{2}\sum_{i \neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{1}{2}\sum_{\alpha\beta} \frac{Z_\alpha Z_\beta}{R_{\alpha\beta}} \qquad \text{(I.3)} \end{aligned}$$

where:

- $\hat{T}_e$ is the kinetic energy of the electrons
- $\hat{V}_{n-e}$ is the potential energy of the electrons in the field of the nuclei
- $\hat{U}_{e-e}$ is the Coulomb interaction between the electrons and
- $\hat{W}_{n-n}$ is the Coulomb interaction between the nuclei

The wavefunction $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$

I.2 Density Functional Theory

A direct solution to the electronic structure problem, this meaning obtaining the ground-state many-body wavefunction $\Psi(\mathbf{r_1}, \ldots, \mathbf{r_N})$ for a given potential is analytically impossible even for a small number of electrons. As such, the need for good approximations to obtain results for real world systems is high. One particularly successful approach is *Density Functional Theory* (DFT).

I.2.1 Hohenberg-Kohn theorems

The basis for DFT lies in the excact reformulation of the outlined electronic structure problem by Hohenberg and Kohn [1]. This reformulation uses the ground state density of the electronic system as the basic variable. [Missing]

Hohenberg-Kohn theorems:

I The external potential (and by extension the ground state wave function and the ground state energy) are unique functionals of the ground state density (except for an additive constant).

II The ground state energy minimizes the energy functional,

$$E[n(r)] > E_0 \; \forall n(r) \neq n_0(r)$$

.

I.2.2 Kohn-Sham equations

One way of approximating the functional $F[n]$ was given by Kohn and Sham [2]. The idea is to use a non-interacting auxiliary system of electrons and introduce a correction potential.

I.3 Parallel computing

The following section will give an overview of the technical aspects of running computer code (such as QUANTUM ESPRESSO) on massively parallel computing environments (such as the PHYSnet compute cluster). The information presented can be found in any textbook on parallel or high-perfomance computing [3].

I.3.1 On scalabilty

In scientific computing, one can identify two distinct reasons to distribute workloads to multiple processors:

- The execution time on a single core is not sufficient. The definition of sufficient is dependent on the specific task and can range from „over lunch“ to „multiple weeks“
- The memory requirements grow outside the capabilities of a single core

In order to judge how well a task can parallelized, usually some sort of scalabilty metric is employed, for example:

- How fast can a problem be solved with N processors instead of one?
- What kind of bigger problem (finer resolution, more particles, etc.) can be solved with N processors?
- How much of the resources is used for solving the problem?

The speedup by using N workers to solve a problem instead of one is defined as $S = \frac{T_1}{T_N}$, where T_1 is the execution time on a single processor and T_N is the execution time on N processors. In the ideal case, where all the work can be perfectly distributed among the processors, all processors need the same time for their respective workloads and don't have to wait for others processors to finish their workload to continue, the execution time on N processors would be $\frac{T_1}{N}$, so the speedup would be $S = \frac{T_1}{\frac{T_1}{N}} = N$.

In reality, there are many factors either limiting or in some cases supporting parallel code scalability. Limiting factors include:

- *Algorithmic limitations*: when parts of a calculation are mutually dependent on each other, the calculation cannot be fully parallelized
- *Bottlenecks*: in any computer system exist resources which are shared between processor cores with limitations on parallel access. This serializes the execution by requiring cores to wait for others to complete the task which uses the shared resources in question
- *Startup Overhead*: introducing parallelization into a programm necessarily introduces an overhead, e.g. for distributing data across all the processors
- *Communication*: often solving a problem requires communication between different cores (e.g. exchange of interim results after a step of the calculation). Communication can be implemented very effectively, but can still introduce a big prize in computation time

On the other hand, faster parallel code execution can come from:

- *Better caching*: when the data the program is working with is distributed among processors (assuming constant problem size), it may enable the data to be stored in faster cache memory. Modern computers typically have three layers of cache memory, with level 1 cache being the smallest and fastest and level 3 being the largest and slowest, so smaller data chunks per processor can lead to the data not being stored in main memory, but completely in cache or in a faster cache level

I.4 Quantum ESPRESSO

II Results

II.1 Parallelisation of self-consistent calculations of electronic-structure properties

II.1.1 First scaling tests

The first step in analysing the scaling of QUANTUM ESPRESSO is to perform a baseline scaling test without any optimisations yet appplied. In Fig. II.1 two scaling tests on the earlier mentioned benchmarking systems Si and TaS2 are pictured. The QUANTUM ESPRESSO version used is compiled using OpenMPI 4.1.0 , without any further compilation or runtime optimisation parameters used.

What exactly? Which compiler?

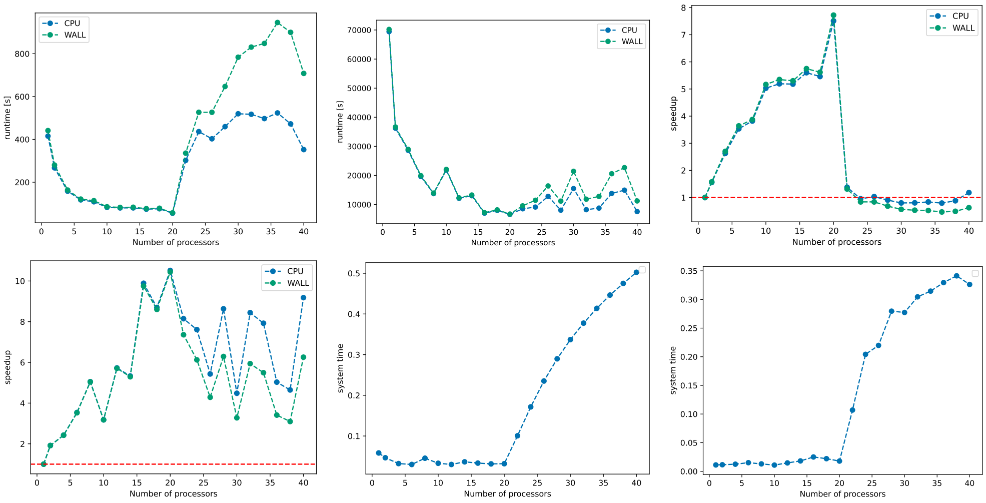

Abbildung II.1: *XXX*

caption

Three different metrics of scalability are pictured in II.1.

- runtime: absolute runtime of the compute job
- speedup: runtime divided by runtime of the job on a single core
- system time: percentage of wall time used by system tasks, e.g. writing to disk, etc.

For further analysis mainly speedup will be used as a metric of scalability, because it lends itself to easy interpretation: optimal scalability is achieved when the speedup scales linearly with the number of processors (with a slope of one), as discussed in ch. I.3.

On a single node, both the Si and TaS2 calculations show good, but not perfect scaling behavior: the speedup does approximately scale linearly with the number of processors, but the slope is round about $\frac{1}{2}$. Even though the scaling behavior is not perfect, there is just a small, almost constant amount of runtime used by system calls, this speaks for good parallelisation: as discussed in ch. I.3, startup time is part of every

When using more than one node, not only does the scaling get worse, the execution needs longer than on a single core for the Si system, with a marginally better perfomance for the TaS2 system. This is also seen in the plots of system time. The percentage of time used for tasks not directly related to calculations (mostly exchange of data in this case, which induces long waiting time when the connection between processors is not as fast as on one motherboard) goes from a near constant value for under 20 processors to 50% of the execution time for the Si system and 35% for the TaS2 system.

These scaling tests pose now two questions to be answered:

- Is better scaling on a single node possible?
- How can scaling over more than one node be achieved?

II.1.2 Different compilers

II.1.3 Using the parallelisation parameters of Quantum ESPRESSO

II.1.4 Comparison with calculations on the HLRN cluster

II.1.5 Scaling over 40 processors

II.1.6 Conclusion: Parameters for optimal scaling

Literatur

[1] P. Hohenberg und W. Kohn. „Inhomogeneous Electron Gas“. In: *Phys. Rev.* 136.3 (Nov. 1964). Publisher: American Physical Society, B864–B871. DOI: `10.1103/PhysRev.136.B864`.

[2] W. Kohn und L. J. Sham. „Self-Consistent Equations Including Exchange and Correlation Effects“. In: *Phys. Rev.* 140.4 (Nov. 1965). Publisher: American Physical Society, A1133–A1138. DOI: `10.1103/PhysRev.140.A1133`.

[3] G. Hager und G. Wellein. *Introduction to High Performance Computing for Scientists and Engineers.* 0. Aufl. CRC Press, 2. Juli 2010. ISBN: 978-1-4398-1193-1. DOI: `10.1201/EBK1439811924`.