Distributed Computing in Molecular Modeling: Practical Experience on the Joint ISMA-ISC Cluster

Oleg Zhikol

SSI STC "Institute for Single Crystals" NAS of Ukraine, 60 Nauki ave., Kharkiv, Ukraine

zhikol@xray.isc.kharkov.com

Abstract. The experience of using a set of molecular modeling programs on the Joint ISMA-ISC computer cluster is discussed. The main attention is paid to the computational efficiency and the scalability of the tasks in different applied program packages. Hardware issues limiting the efficiency are pointed out. Specific features of some quantum chemical modeling programs are mentioned briefly.

Keywords

Distributed computing, quantum chemistry, molecular dynamics, computation efficiency, parallelization libraries, joint ISMA-ISC cluster.

1 Introduction

Growth of the computing power has been quite intensive few later tens of years. At the same time, molecular modeling had becoming an extensive and valuable part of the contemporary science. A much cheaper alternative to supercomputers represents Beowulf-type clusters. Our Institute for Scintillation Materials – STC "Institute for Single Crystals" Joint computer cluster begun as one of such in 2000. Now it is just a few steps closer to a professional system, with total 400 CPU cores, 100 Mb Ethernet and partly Infiniband interconnect, and a few shared RAID arrays.

Here we limit discussion of the immense molecular modeling field to the tasks actually being solved on our cluster. Our main specialization is the quantum chemistry (QC). Its methods are accessed through the following program packages: Gaussian'09 [1], GAMESS US [2], NWChem [3], Orca [4], plus few rarely-used ones. The other task type is the molecular dynamics (MD), both classical MD within GROMACS package [5] and ab initio MD within CPMD one [6].

We will review briefly the computational requirements of the mentioned problems and discuss specifics of the using distributed computing to solve them. We address also briefly the cluster configuration and statistics, as well as cluster connection to the Ukrainian National Grid.

2 Related works

Perspectives of molecular modeling on either supercomputers or computer clusters are evident and barely deserve special discussions [7]. The field is still growing rapidly, while yearly relative gain is decreasing now. The trend is supported by the data on the amount of computational power, consumed by our users during the later 5 years, see table 1. Here we omit the amount of CPU time consumed by international collaborations in the physics field (ALICE, Virgo VO in the European Grid Infrastructure).

<table>
<thead>
<tr>
<th>Year</th>
<th>2013</th>
<th>2014</th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time,×10^4 hrs</td>
<td>625.3</td>
<td>483.7</td>
<td>673.3</td>
<td>740.7</td>
<td>806.9</td>
</tr>
</tbody>
</table>

An accent is shifting slowly from QC modeling to MD simulations. More exact estimation is unavailable since many MD studies are carried privately by pharmacy-related companies, with results either not announced publicly or, at least, presented delayed for several years.

We would like to discuss here our progress made under present conditions, namely, with no strong financial support.
3 Main sections

The molecular modeling tasks are very diverse in their nature, in the used algorithms and their implementations. It holds true even under our limitation by practically solved tasks on the Joint ISMA-ISC cluster. We consider below the two main routes separately.

3.1 Molecular dynamics simulation tasks

The simulation handles coordinates and velocities of the atoms. The system evolution in time is defined either by classical potential, inexpensively computed, or by ab initio potential computed usually in the Car-Parrinello approach [8]. Due to an infinite size of the system modeled, plane waves and Fourier transform are used extensively.

The parallel program threads require access to the whole set of the data. Such computation remains extremely efficient on the shared memory, while under distributed computing the interprocess communication time becomes crucial rapidly. The data volume to be broadcasted by a thread is relatively small, about 7A/C single words, where A – number of atoms, C – number of threads. In our tasks, typical values are 10⁷…10⁸ for A, and C below 64. On the other hand, an MD run requires 10⁴…10⁹ such steps. The interconnect network latency is essential. The best hardware configuration we could get at that time was a cluster part of 15 nodes each with 8 CPU cores (two Xeons E5345@2.33 GHz) on Infiniband 24-port switch. The part is dedicated to the MD and similar network latency-sensitive tasks.

The GROMACS software performance (classical MD) scales in this environment close to linearly up to 64 CPU cores (8 nodes), with 98-99% efficiency at 24 cores (3 nodes). We define the efficiency term as the CPU time to wall time ratio.

The GROMACS software performance (classical MD) scales in this environment close to linearly up to 64 CPU cores (8 nodes), with 98-99% efficiency at 24 cores (3 nodes). We define the efficiency term as the CPU time to wall time ratio. Ab initio MD must scale even better since the computation itself is more CPU intensive, but we lack enough CPMD usage statistics to draw reliable conclusions.

3.2 Quantum chemical modeling tasks

The sought-for wavefunction is constructed as a linear combination of atom-centered gaussian-like (basis) functions. It holds usually for the total number of basis functions \(B = 20A\) (like above, \(A\) is the number of atoms). The main source of computational complexity are so-called two-electron integrals dependent each on 4 basis functions. Number of the integrals is estimated as \(B^2\) with a prefactor under 1. Thus, a typical task with \(A \approx 30\) gives about \(10^{15}\) integral entries at least 3 single words long each, i.e. at least 1.5 TB data in total. While the file should be saved once and read several times by all threads in sequential mode, its size may be unacceptably high.

In view of it, all contemporary QC programs provide the "direct" integral computation mode, when the integrals are recomputed at need without saving them to disk. The calculation is repeated 15-30 times and, without specific hardware, remains more efficient than the "conventional" method above. It also scales well with the number of CPU cores used, even at relatively slow interconnect network like ours.

More exact QC approximations include four-index transformation of the integral set. The transformed integral file would be saved once in sequential mode and read several times in random access mode. The access mode makes high demands to either data storage device or interprocess communication speed, for replicated or distributed data storage type, respectively. Unfortunately we cannot dedicate a data server with large and fast enough storage to the purpose. With such a server, one could join up to 16 cluster nodes even with gigabit Ethernet to reach a configuration appropriate for the four-index transformation with the data set saved. Note that a try to use the same bandwidth and a part of the node local disk space for a distributed file system was unsuccessful: a single task might cause a great drop of the whole cluster performance. The only successful application to a large enough system \((A \approx 70, B \approx 1300)\) we've got years ago on the Beowulf-type hardware: 2Gb RAM single-CPU Athlon XP 2 GHz nodes, connected by 100 Mb Ethernet, – and only using a specific program package, namely NWChem. That hardware handled the task with a reasonable (above 80%) efficiency for 8-12 nodes.

This "conventional" mode of the four-index transformation is used only for small model systems, whenever most of the data sets can be cached in RAM. All QC programs have some direct or semidirect alternative with recomputing some parts of the data sets at need. The alternatives do not scale well at 100 Mb / 1 Gb Ethernet and require considerable amount of the local storage space still.

For many QC calculation types, the integral derivatives over atomic coordinates are necessary. Usually these data are calculated just once for given atom positions, and, in spite of a huge data set size, the only issue is the availability of the corresponding functionality in the QC programs. For instance, the second derivatives are available mostly in Gaussian'09, which we use only on shared memory.
Thus, our primary cluster configuration (nodes with Intel E3-1240 @ 3.4 GHz CPU, 16 Gb RAM, 512 Gb local disk on the 100 Mb Ethernet) is suitable for solving many QC tasks in the lowest approximations (self-consistent field, including density functional theory) and is partially usable for the tasks requiring the four-index transformation. In the later case, the distributed data storing scheme allows us to carry calculation out with a substantially degraded (50% and down to 5-10%) efficiency.

3.3 On the implementation of parallelism in the QC modeling programs

The operations to be parallelized belong mostly to the linear algebra domain: building and transforming the matrices. The parallel QC programs use SIMD model usually. The underlying message passing layer is provided by an MPI implementation. The most used implementations are OpenMPI [9] and Intel MPI [10]. Note that, for historical reasons, not every each modeling program depends on MPI. For example, TCP Linda parallelization in Gaussian [11] is an interface created before any MPI standard manifestation. We have no Linda experience due to financial limitations, using parallel Gaussian in shared memory mode only. Similarly, GAMESS US can optionally use an own TCP-based message passing layer, and NWChem can use the TCGMSG library [12]. The recommended way for these two programs is relying on an MPI.

The NWChem QC program [3] is worth a discussion here. Being relatively new, it had been developed with high portability and parallel execution efficiency in mind. The parallelized code is well structured, relying on the Global Arrays (GA) tool [13] by the same developer. The tool implements array handling and linear algebra operations in parallel, depending in its turn on the widely known libraries BLAS and LAPACK. A special attention is paid to the interprocess communication efficiency [14], the implemented algorithms are aimed at that as well. From the developer point of view, the GA provides very convenient interface for memory management and matrix operations; the interface remains tractable even in its FORTRAN version. The common tasks in the program scale very good up to the maximum tested 64 cores in our 4x16 nodes variant, when the acceleration coefficient tends to 60. The low comparison base, i.e. low single-CPU performance, is leveled at such considerable number of threads. The drawback is somewhat limited functionality of the program itself, in particular, not implemented integral second derivatives.

Another example of a highly-parallelized modeling program delivers GROMACS classical MD package [5]. In contrast, the most others programs used by us are parallelized just partly, with lots of replicated computations.

3.4 Setup of the Joint ISMA-ISC cluster

The Joint ISMA-ISC cluster was set up basing on the free cluster management software. The main components are the following:

- PERCEUS provisioning manager [15], providing a uniform boot scheme for the nodes,
- Torque [16], a PBS-based resource manager, connecting the task pool to the resource pool,
- Maui scheduler [17], deciding scheduled task priorities.

This software allowed also integration of the cluster into the Ukrainian National Grid [18]. The corresponding Tier-3 site is known in the EGI as ISMA-UA.

4 Conclusion

Distributed computing provides to date a much cheaper but still reliable alternative to the supercomputers in the molecular modeling field, though some specific methods may require a specific hardware.

Despite that our hardware is about 10 years old, it is actual and adequate enough to the molecular modeling tasks not only of the SSI STC “Institute of Single Crystals”, but of about fifteen institutions of the National Academy of Science and of the Ministry of Education and Science of Ukraine as well. At the same time it functions as a part of the shared resource center “Molecular and crystal material structure” [19] and as a part of the Ukrainian National Grid.

5 Acknowledgments

The maintaining and development of the Joint ISMA-ISC cluster is partially supported by the projects in the frame of the Scientific research program by the NAS of Ukraine “Grid infrastructure and grid technologies for scientific and application tasks”, order No. 212 by 16.04.2018, project No.16.
References

[18] http://ung.in.ua