Average Vibrational Potentials of Oscillators in Condensed-matter Environments using Hadoop

Bojana Koteska, Anastas Mishev
Faculty of Computer
Science and Engineering,
Ss. Cyril and Methodius University,
Rugjer Boshkovikj 16, P.O. Box 393,
1000 Skopje, Republic of Macedonia
Email: {bojana.koteska, anastas.mishev}@finki.ukim.mk

Ljupčo Pejov
Institute of Chemistry,
Faculty of Natural Sciences
and Mathematics,
Ss. Cyril and Methodius University,
Arhimedova 5, P.O. Box 162,
1000 Skopje, Republic of Macedonia
Email: ljupcop@iunona.pmf.ukim.edu.mk

***Abstract*—In physical sciences, when condensed matter systems (e.g. solids or liquids) are modeled with an explicit inclusion of dynamical effects, often the following computational problem arises. A given property of an embedded atomic/molecular system within condensed phase should be computed either at different possible structural arrangements and further average over configurations, or alternatively, it is possible to generate an averaged configuration of the dynamical surrounding that the system experiences and further compute the property of interest at that configuration. The problem of solving the average vibrational potentials of large number of oscillators in various condensed-matter environments (sampled from a statistical physics simulation) can be placed in the category of problems with large data sets. In this paper, a distributed and parallel processing of the large data sets needed for the generation of the averaged vibrational potential is efficiently performed by using the MapReduce programming model and Hadoop software library. Some of the reasons for choosing the Hadoop software library are: It is able to work on data pieces in parallel; The computing solutions enabled by Hadoop are scalable and flexible; The distributed file system enables rapid data transfer among nodes; Hadoop is fault-tolerant which means that if a node fails the job is redirected to another node. The main goal of this paper is to perform an efficient processing of the large data sets used in the scientific applications.**

***Index Terms*—Hadoop, Average vibrational potentials, Anharmonic oscillator, Condensed-matter environments, Schrödinger equation**

I. Introduction

Theoretical models in physical sciences are often used to understand the experimentally observed behavior of certain physical systems or to predict their behavior under specific circumstances which are relevant to the actual or potential technological applications of the systems in question. Besides getting a more enlightening view of the systems behavior, theoretical models may be quite useful in discriminating among various factors leading to observation of certain physical phenomena or in quantifying the contribution of various factors to a certain physical observable. Most of the experimental data are, however, collected at finite temperatures, usually quite above absolute zero.

A reliable theoretical model aiming to provide a realistic description of the system in question therefore has to account for the dynamical effects on a certain time-scale. Most of the models based on quantum mechanical description of many-particle physical systems are based on explorations of the potential energy hypersurfaces (or certain cuts through these surfaces), which means that they do not conform to the previously mentioned criterion. To explicitly include the dynamical behavior of the studied quantum system, one has to treat it within the framework of quantum dynamics. However, a fully exact quantum dynamical treatment of multi-particle systems is prohibitively computationally expensive. At the same time, luckily, such full quantum dynamical treatment is mandatory only in certain specific cases, usually when the focus of the study is put on light particles (such as e.g. hydrogen atoms).

An acceptable alternative which has been exploited to some extent in the literature is to first carry out a classical dynamics (or statistical physics, such as e.g. Monte Carlo) simulation of the time-evolution (or evolution in imaginary time) of the system in question, then to pick up a reasonably small number of configurations (snapshots from the classical simulation) and perform rigorous quantum mechanical simulations only on these configurations. Though the previously mentioned dynamical simulations are classical in a rigorous sense, note that the interaction potentials used throughout the simulations may be even derived from high-level quantum mechanical calculations.

II. Related work

There are several papers in which MapReduce paradigm has been used for solving problems in the scientific domain. In [1], the authors applied MapReduce model to perform High Energy Physics data analyses and Kmeans clustering. They also made a streaming-based MapReduce implementation and compared its performance with Hadoop. Their conclusion is that most of the scientific analyses that has some form of the SMPD algorithm can benefit from the MapReduce model and can achieve scalability and speedup.

In [2], the authors present the MapReduce implementation in Google inc. The implementation is highly scalable and it processes terabytes of data on thousands of machines. Also, upwards of one thousand MapReduce jobs are executed on

Google's clusters every day. MapReduce model is used for sorting, data mining, machine learning, generation of the data of the web server, etc.

In his thesis [3], the author propose a novel solution for molecular dynamics simulation based on Hadoop MapReduce. The solution can predict the execution time of a given size molecular dynamics simulation system. He also presents the performance and energy consumption improvement of the solution which is implemented in a hybrid MapReduce environment.

Bunch et al. [4] explore which scientific computing problems can be solved by using MapReduce and which can not. They implement different non-trivial algorithms with MapReduce and measure their performance. The authors found out that the MapReduce framework is not suitable for iterative algorithms where each iteration runs a number of MapReduce jobs.

In their paper [5], the authors propose architecture for a configuration implemented in a scientific private cloud prototype and they use Hadoop to achieve scalability and fault tolerance. The experiments showed the effectiveness of the proposed model. In [6], the authors describe the development of the Hadoop-based cloud scientific computing application that processes sequences of microscope images of live cells.

A Hadoop plugin that allows scientists to specify logical queries over array-based data models is presented in [7]. It executes queries as MapReduce programs defined over the logical data model. The goal of this paper is to reduce total data transfers, remote reads and unnecessary reads.

III. Average Vibrational Potentials of Oscillators in Condensed-matter Environments

In physical sciences, when condensed matter systems (e.g. solids or liquids) are modeled with an explicit inclusion of dynamical effects, often the following computational problem arises. A given property of an embedded atomic/molecular system within condensed phase should be computed either at different possible structural arrangements and further average over configurations, or alternatively, it is possible to generate an averaged configuration of the dynamical surrounding that the system experiences and further compute the property of interest at that configuration.

For example, if one is interested in an anharmonic oscillator embedded in a solid or liquid, the vibrational potential of the form:

$$V(r) = V_0 + 1/2k_2r^2 + k_3r^3 + k_4r^4 + k_5r^5 \qquad (1)$$

may be computed at n configurations and then the vibrational Schrödinger equation solved for each particular $V_i(r)$. In previous equation, r is an appropriately chosen vibrational coordinate, k_2 is the harmonic force constant, while k_3, k_4 and k_5 are cubic, quartic and quintic anharmonic force constants respectively [8][9]. In these papers this approach and generated the vibrational density of states for a number of X-H oscillators embedded in a variety of liquid environments have been exploited. Though such approach is computationally feasible, in some cases, especially if one is interested only in the average frequency (or frequency shift), it would be desirable to avoid explicit computation of vibrational frequencies by solving the vibrational Schrödinger equation for all $V_i(r)$. Instead, one could use a single computation of this type, for an averaged configuration or averaged potential within the condensed phase medium. In the present study, we further elaborate the previous two ideas, by considering the averaged vibrational potential instead.

Alternatively to the averaged configuration or averaged environmental potential approaches, one can generate an averaged vibrational potential of the form:

$$<V(r)>=<V_0> +1/2 <k_2> r^2+ <k_3> r^3+ \\ <k_4> r^4+ <k_5> r^5 \qquad (2)$$

(where $<>$ denotes ensemble averaging or averaging over time configurations) and subsequently solve the vibrational Schrödinger equation for such averaged potential energy function. To illustrate the concept and consider a particular physical system, we consider the fluoroform-dimethylether dimer embedded in liquid krypton, which has been a subject of attention in our recent paper. The main interest for this system, which has previously been studied by cryospectroscopic techniques, is driven by the peculiar behavior of the C-H vibrational mode of the fluoroform moiety upon complexation with dimethylether, that exhibits C-H stretching frequency blue shift (instead of the expected red shift by "chemical intuition"). The details concerning the mechanism behind the blue shift and many other aspects in this context have been discussed in details in our previous work.

In the present paper, we focus on the development of method, based on the map-reduce computational approach, to extract the "solvent-averaged" X-H stretching vibrational potential. We have therefore computed the vibrational potential energy functions for at least 50 C-H stretching oscillators of the CF_3H moiety within the CF_3H - $(CH_3)_2O$ dimer at B3LYP, HF and MP2 levels of theory. The 6-31++G(d,p) basis set has been used for orbital expansion in all calculations. The positions of the C and H atoms in the course of "excitation" of the C-H stretching vibration have been generated by fixing the center-of-mass of the C-H bond fixed, as explained in details elsewhere. 20-point grids were used to scan the C-H stretching potential energy function, spanning a suitable range of C-H distances, so that the potential is sampled in the areas in which the wavefunctions corresponding to the ground and the first excited vibrational states are already decayed to zero. The data generated in such way were further interpolated by a fifth-order polynomial in the C-H distance r (Eq. 1) . The functions of the form Eq. 1 were further cut after the fourth-order term, transformed into Simons-Parr-Finlan type coordinates $\rho = r - r_e/r$ (where r_e is the equilibrium value of the C-H distance), and the vibrational Schrödinger equation was solved by the variational method (the linear variant). For that purpose, harmonic oscillator eigenfunctions were used as an orthonormal basis set. To generate the averaged potential of the

form Eq.2 by the map-reduce technique, we have averaged the values of molecular potential energies (in Born-Oppenheimer sense) at each r(C-H) value. The resulting average vibrational potential energy function of the form Eq.2 was further also cut after fourth order, transformed into SPF-type coordinates, and subsequently the vibrational Schrödinger equation was solved in a variational manner. The map-reduce approach, as implemented in Hadoop, was used as explained below.

IV. The MapReduce Model

MapReduce is a programming model for processing large data sets in parallel [10]. The partitioning of the input data and the scheduling of the program's execution across multiple machines are responsibilities of the run-time system. The user should specify a map function (mapper) which processes key-value pairs and a reduce function (reducer) which merges all the intermediate values associated with the same intermediate key [2].

The MapReduce model can be divided into rounds, each containing three phases: *Map*, *Shuffle and Sort* and *Reduce*. The *Map* phase maps each single pair of (key, value) to the machines in the run-time system as a new multiset of (key,value) pairs where the value in each new pair is a substring of the original value. The *Shuffle* phase is responsible for sorting and transferring the map outputs to the reducers. The *Reduce* phase computes some function on the data on each machine [11].

A MapReduce program consists of finite sequence of rounds specified as 2-tuples (tuples of two elements), each tuple containing a map and a reduce function. Formally, this can be written as: $((M_1, R_1), (M_2, R_2), ..., (M_n, R_n))$ where M_i is a mapper, R_i is a reducer, i is an integer number and $1 \leq i \leq n$. A 2-tuple is defined as (M_i, R_i). Let the program input that is a multiset of (key;value) pairs be denoted by U_0 and the output that is a multiset of (key;value) pairs of the i-th round by U_i.

The program executes for $r = 1, ...n$. For each r, the *Map*, *Shuffle* and *Reduce* phase are performed. The *Map* phase feeds each (key;value) pair $(k; v)$ in U_{r-1} to the mapper M_r and runs it. The output of the mapper M_r will be a sequence of (key;value) pairs $(k_1; v_1)$, $(k_2; v_2)$,... and it can be defined as: $U'_r = \cup_{(k;v)\in U_{r-1}} M_r((k; v))$. The *Shuffle* phase constructs $V_{k,r}$ (values such that $(k; v_i) \in U'_r$) from U'_r for each k. The *Reduce* phase feeds the k and some arbitrary permutation of $V_{k;r}$ to the separate instance of the reducer R_r and runs it for each k. The output of the reducer is a sequence of 2-tuples $(k; v'_1)$, $(k; v'_2)$,... and U_r that is a multiset of (key;value) pairs produced by the reducer R_r is defined as $U_r = \cup_k R_r((k; V_{k,r}))$ [12] [13].

Programs that use the MapReduce model implement the Mapper and Reducer interfaces to provide the map and reduce functions. The map and reduce methods can be represented as shown below. The values with the same key are reduced together [14].

method Map(key k, value v) $\rightarrow$ EMIT(key k', value v')

method Reduce(key k, value v) $\rightarrow$ EMIT (key k', value $[v', v_2, v_3...]$

The MapReduce model automatically supports parallel programming and it shields the programmer from writing code about data distribution, scheduling and fault tolerance. The programmer should only specify the map and reduce functions. This also can be considered as a disadvantage of the model since the programmer cannot affect the efficiency of the parallelism. Thus, it is not always clear which kind of problems are suitable to be solved using the MapReduce model and which not [13]. The scientific data volumes and clustering algorithms used in chemistry, biology, physics are computing intensive operations and the use of parallelization techniques is key in order to achieve efficient data analyzes. MapReduce model is suitable when processing of the data should be split into smaller independent computations and the intermediate results should be merged after some post-processing in order to get the final result. It provides simplicity, robustness and has less synchronization constraints which supersede the additional overheads [1].

Hadoop (Apache Hadoop) is an open source software data-processing library which allows distributed and parallel processing of large data sets. The Hadoop project includes four different modules: Hadoop Common (utilities that support the other Hadoop modules), Hadoop Distributed File System (distributed file system that provides high-throughput access to data), Hadoop YARN (framework for job scheduling and cluster resource management) and Hadoop MapReduce (A YARN-based system for parallel processing of large data sets) and it is used by many companies including Facebook, Cloudera, Amazon, Microsoft, Yahoo, etc. The data processing in Hadoop can be implemented in MapReduce directly or by using high-level languages and translating into Map-reduce jobs later [15][16]. Hadoop can be also used for building data warehousing solutions. An example is Hive which supports queries expressed in a HiveQL (SQL-like declarative language). The queries are compiled into map-reduce jobs and executed on Hadoop [17].

V. Computing Averaged Vibrational Potentials Energies by using Hadoop

The purpose of the our algorithm is to compute the average vibrational potential energies for at least 50 C-H stretching oscillators of the CF_3H moiety within the CF_3H - $(CH_3)_2O$ dimer at B3LYP, HF and MP2 levels of theory. Calculating the average values, in our case the average vibrational potential energies, is a typical map-reduce problem. Each document that describes the same C-H stretching oscillator in a different environment has two columns, one containing the distances r(C-H) and the other containing values of molecular potential energies (U). The pseudo code of the Map and Reduce methods used in our algorithm is given below.

```
Method Map(String r, String U):
// r: input key r(C-H)
// U: input_value
```

```
for each r in all documents:
EmitIntermidiate(r,ParseDouble(U));

Method Reduce(String r, Iterator interm_vals):
// r: key, same as input_key
// interm_vals: intermediate values-
// list of all U-s group by r
double sum=0,result=0;
for each v in interm_vals:
sum += v;
result=sum/length(interm_vals);
Emit(AsString(result));
```

The algorithm is implemented in Java and it was performed three times, once for each level of theory (B3LYP, HF and MP2).

The main results from the present study are summarized in Fig. 1 a-c. In Fig. 1, the vibrational density-of-states (DOS) histograms generated from the computed vibrational frequencies of the $|0> \rightarrow |1>$ C-H stretching vibrational transition are presented, together with the delta-like function (with dashed lines) representing the frequency of the $|0> \rightarrow |1>$ transition obtained for an averaged C-H stretching potential. In the same figure, also the numerical values of the corresponding frequencies are given. As can be seen, if one is interested solely in the average vibrational frequencies, and not in the corresponding distributions, our "averaged vibrational potential" approach gives excellent results. The matching between average vibrational frequencies computed from the DOS distributions, and the single vibrational frequency values computed from only a single averaged vibrational potentials is excellent, regardless on the level of theory. No biasing effects are present.

VI. Conclusion

In this paper we have presented the benefits of using the MapReduce model and Hadoop framework in scientific domains. The average vibrational potentials of oscillators in condensed-matter environments were computed only by specifying map and reduce functions implemented in Hadoop. The vibrational potential energy functions were computed for at least 50 C-H stretching oscillators of the CF_3H moiety within the CF_3H - $(CH_3)_2O$ dimer at B3LYP, HF and MP2 levels of theory. The results show that there is an excellent matching between average vibrational frequencies computed from the DOS distributions, and the single vibrational frequency values computed from only a single averaged vibrational potentials. By using the results, the vibrational Schrödinger equation was solved in a variational manner. Since there are many big-data oriented problems in the scientific domains, the MapReduce paradigm and Hadoop can be suitable for their solving, especially when some reduction of the data should be performed. Our future work is oriented to solving more difficult problems in the scientific domains by using the MapReduce method and Hadoop framework.

References

[1] J. Ekanayake, S. Pallickara, and G. Fox, "Mapreduce for data intensive scientific analyses," in *Proceedings of the 2008 Fourth IEEE International Conference on eScience*, ser. ESCIENCE '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 277–284. [Online]. Available: http://dx.doi.org/10.1109/eScience.2008.59

[2] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[3] C. He, "Molecular dynamics simulation based on hadoop mapreduce," Ph.D. dissertation, Computer Science and Engineering, Department of University of Nebraska-Lincoln, Lincoln, Nebraska, May 2011.

[4] C. Bunch, B. Drawert, and M. Norman, "MapScale: A Cloud Environment for Scientific Computing," University of California, Tech. Rep., Jun. 2009. [Online]. Available: http://www.google.ch/search?q=Future+of+MapReduce+for+scientific+computing&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:de:official&client=firefox-a

[5] Y. Tabaa and A. Medouri, "Towards a next generation of scientific computing in the cloud," *International Journal of Computer Science Issues*, vol. 9, no. 3, November 2012.

[6] C. Zhang, H. Sterck, A. Aboulnaga, H. Djambazian, and R. Sladek, "Case study of scientific data processing on a cloud using hadoop," in *High Performance Computing Systems and Applications*, ser. Lecture Notes in Computer Science, D. Mewhort, N. Cann, G. Slater, and T. Naughton, Eds. Springer Berlin Heidelberg, 2010, vol. 5976, pp. 400–415. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12659-8_29

[7] J. B. Buck, N. Watkins, J. LeFevre, K. Ioannidou, C. Maltzahn, N. Polyzotis, and S. Brandt, "Scihadoop: Array-based query processing in hadoop," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '11. New York, NY, USA: ACM, 2011, pp. 66:1–66:11. [Online]. Available: http://doi.acm.org/10.1145/2063384.2063473

[8] E. Kohls, A. Mishev, and L. Pejov, "Solvation of fluoroform and fluoroformdimethylether dimer in liquid krypton: A theoretical cryospectroscopic study," *The Journal of Chemical Physics*, vol. 139, no. 5, pp. –, 2013. [Online]. Available: http://scitation.aip.org/content/aip/journal/jcp/139/5/10.1063/1.4816282

[9] V. Kocevski and L. Pejov, "Anharmonic vibrational frequency shifts upon interaction of phenol(+) with the open shell ligand o2. the performance of dft methods versus mp2," *The Journal of Physical Chemistry A*, vol. 116, no. 8, pp. 1939–1949, 2012. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/jp209801s

[10] R. Lmmel, "Googles mapreduce programming model revisited," *Science of Computer Programming*, vol. 70, no. 1, pp. 1 – 30, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167642307001281

[11] T. White, *Hadoop: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2009.

[12] T. Spangler, "Algorithms for grid graphs in the mapreduce model," Master's thesis, University of Nebraska-Lincoln, 2013.

[13] H. Karloff, S. Suri, and S. Vassilvitskii, "A model of computation for mapreduce," in *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '10. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2010, pp. 938–948. [Online]. Available: http://dl.acm.org/citation.cfm?id=1873601.1873677

[14] D. Licari, "Mapreduce," November 2010.

[15] G. Wang, "Evaluating mapreduce system performance: A simulation approach," Ph.D. dissertation, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA, August 2012.

[16] T. A. S. Foundation. Apache hadoop. [Online]. Available: http://hadoop.apache.org/

[17] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive: A warehousing solution over a map-reduce framework," *Proc. VLDB Endow.*, vol. 2, no. 2, pp. 1626–1629, Aug. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1687553.1687609

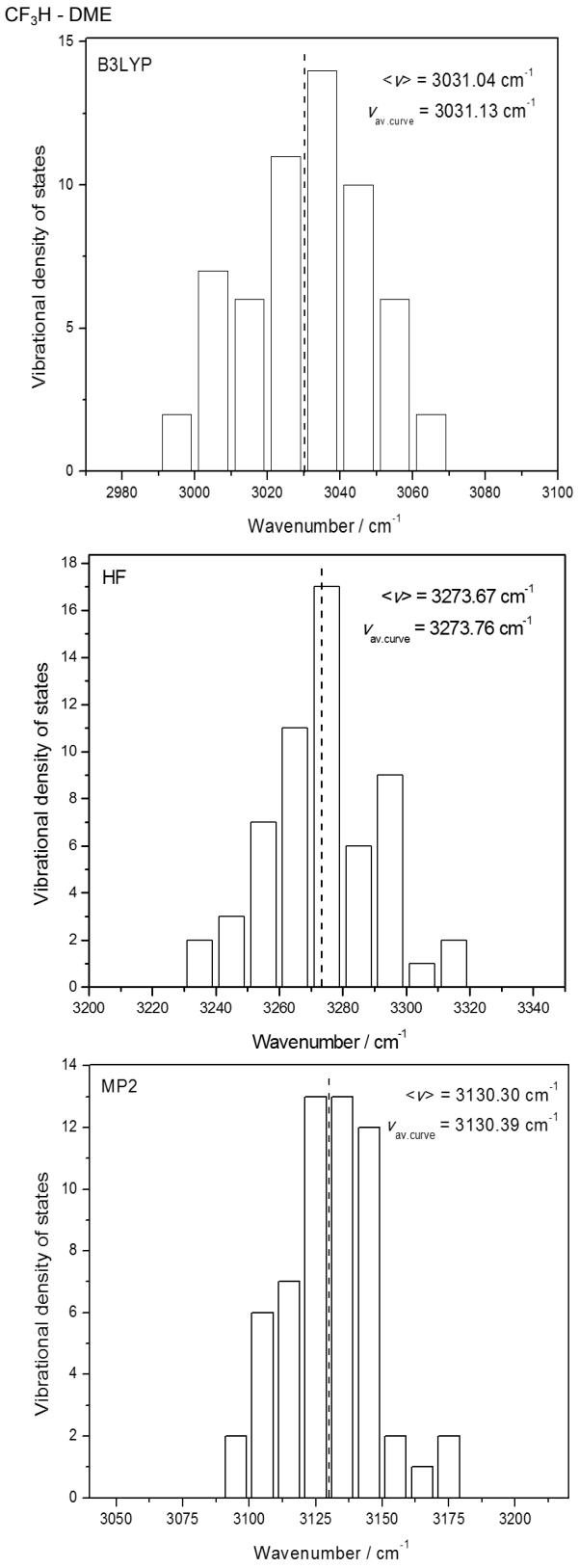

Fig. 1. Vibrational Density-of-states (DOS) Histograms Generated from the Computed Vibrational Frequencies together with the Delta-like Function