Rostec
Techpriemka

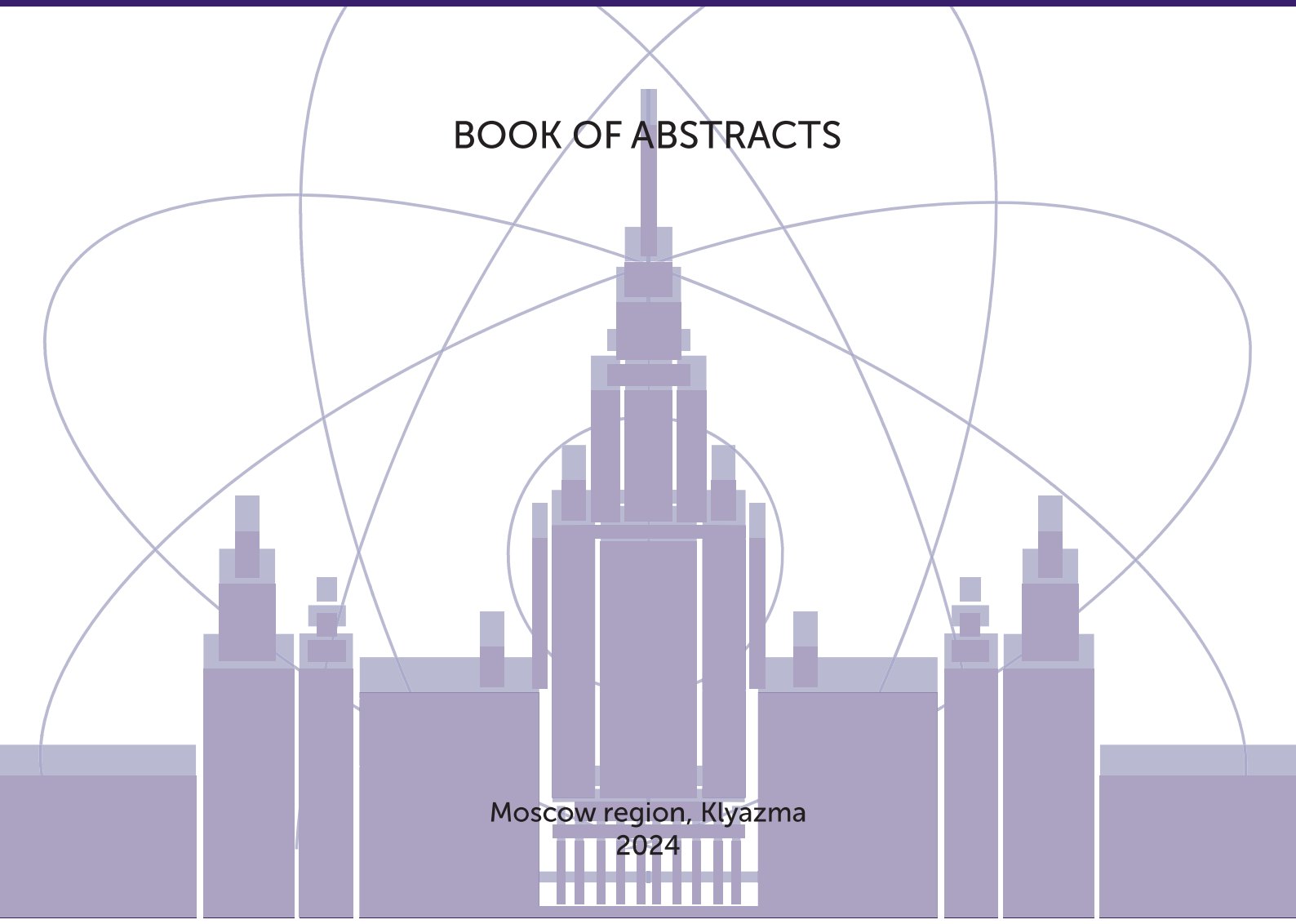
MSU SCHOOL OF ADVANCED ENGINEERING STUDIES

2024 PhysChem

CHALLENGES 2024

INTERNATIONAL SCHOOL AND CONFERENCE "CURRENT CHALLENGES IN CHEMICAL PHYSICS AND THEORETICAL CHEMISTRY"

BOOK OF ABSTRACTS



Moscow region, Klyazma
2024

CHALLENGES 2024

INTERNATIONAL SCHOOL AND CONFERENCE "CURRENT CHALLENGES IN CHEMICAL PHYSICS AND THEORETICAL CHEMISTRY"

Book of abstracts

Moscow region, Klyazma
2024

УДК 544
ББК 24.5я43
М34

Отв. редакторы: А. В. Боченкова, К. В. Казаков

М34 **Материалы Международной школы-конференции по современным проблемам химической физики и теоретической химии «CHALLENGES 2024».** — М.: Издательство "Перо", 2024. — 120 с.

ISBN 978-5-00244-253-9

Сборник включает в себя программу и аннотации докладов, представленных на Международной школе-конференции по современным проблемам химической физики и теоретической химии «CHALLENGES 2024» (пос. Поведники, Московская область, 1-5 июля 2024 г.). Тематика школы-конференции охватывает следующие направления фундаментальных исследований:

- Современные методы квантовой химии;
- Высокопроизводительные вычисления и алгоритмы машинного обучения;
- Молекулярная спектроскопия и физика планетных атмосфер;
- Радиационные аспекты моделирования климата и дистанционного зондирования; природных и антропогенных процессов;
- Фотофизика и фотохимия молекул при линейном и нелинейном возбуждении.

Сборник представляет интерес для специалистов в области физики и квантовой химии.

УДК 544
ББК 24.2я43

ISBN 978-5-00244-253-9

Contents

Assessing containerisation overhead for running Firefly quantum chemistry program

A.A. Moskovsky[1,2] ***S.S. Konyukhov***[2]

[1] M.V. Lomonosov Moscow State University Chemistry Department, Laboratory of Quantum Chemistry and Molecular Modelling
[2] RSC Group

Containerisation technology is widely used nowadays in various software stacks including those require high performance from underlying hardware platforms [1, 2]. The Artificial Intellect (AI) and Deep Learning (DL) may be the most vivid examples. The Firefly quantum chemistry program is quite popular in the theoretical chemists community with thousands of users around the world. Unfortunately, Firefly is is notoriously difficult to install, as highly-optimized kernel require legacy 32-libraries to work. We hope that containerisation approach can be beneficial to Firefly quantum chemistry program, but the related overhead is an open question. Thus in this work we compare execution time of Firefly for test jobs in both native and containerised environments. It was used computational server with following configuration:

- CPU 2x Intel(R) Xeon(R) CPU E5-2603 v4@ 1.70GHz w/o hyperthreading
- memory DDR4 2133 MHz 128 GB;
- server board Intel(R) S2600WT2R;
- hypervisor VMware ESXi™, client version 1.33.4, ESXi version 6.5.0.

Containerised environment was obtained by Docker containerization system running on 1 or 2 virtual machines (denoted further as vnodes) under Rocky Linux 9 operating system government. The total number of containers within vnodes was 0 for native and 1/2/4/8 for containerised mode. In test series with 2 vnodes, containers were evenly distributed between them. Our Docker image was based on Rocky Linux 9 Minimal image (rockylinux:9.0-minimal) and NLKNguyen:alpine-mpich image [3] from the public Docker repository [4].

Table 1: Mean runtime for tests with fixed input size.

User+System Time, sec				
		Native	Containerization	Delta
1 vnode	1 MPI-process	73.9 (0.4)	73.8 (0.5)	-0.1
	2 MPI-processes	43.4 (1.4)	42.0 [0.2]	-1.4
	4 MPI-processes	26.5 (1.0)	25 [2.2]	-1.5
	8 MPI-processes	19.5 (1.2)	17.4 [1.9]	-2.1
2 vnodes	2 MPI-processes	42.8 (1.0)	41.7 (0.8)	-1.1
	4 MPI-processes	30.0 (1.4)	27.75 [0.03]	-2.25
	8 MPI-processes	25.1 (1.7)	24 [2.5]	-1.1
Notifications: $(\dots)$ - standard deviation, sd; $[\dots] = \max - \min \approx 2sd$				

To run our test calculations we used Firefly quantum chemistry software(version 8.2.0/mpich1 for Linux, 32-bit) [5]. As test job we chose optimization of geometry configuration of n-propane molecule by quadratic gradient descent method and DFT/B3LYP5 approximation (basis set 6-31G*, 61 basis set functions). For one series of test runs we fixed the number of optimization steps to 12 and increased it proportionally with a number of MPI-processes in another series (12/24/48/96 optimization steps for 1/2/4/8 MPI-processes, respectively). During test runs all MPI-processes were evenly distributed among containers. We run 32 tests for each meta-parameters combination

such as nodes/containers/MPI-processes. Then we used average numbers for analysis. Time measurements were done with the help of strace profiler and duplicated by Firefly outputs (wall time). Usage of strace profiler functionality allowed us to evaluate the influence of containerization on various quantities which describe runtime process e.g. user and system runtimes or number of system calls. Moreover, our tests show that profiler overhead was negligibly small in comparison to the variance of job execution times. Test job execution times for fixed numerical complexity i.e. for fixed number of optimization steps are presented in the table below. On the Figure 1 we show the graph of execution time respectfully to MPI-process number for two modes (containerised and native), when number of optimization steps increased proportionally to MPI-processes number (weak scaling). The comparison of various quantities, such as mean execution time, standard

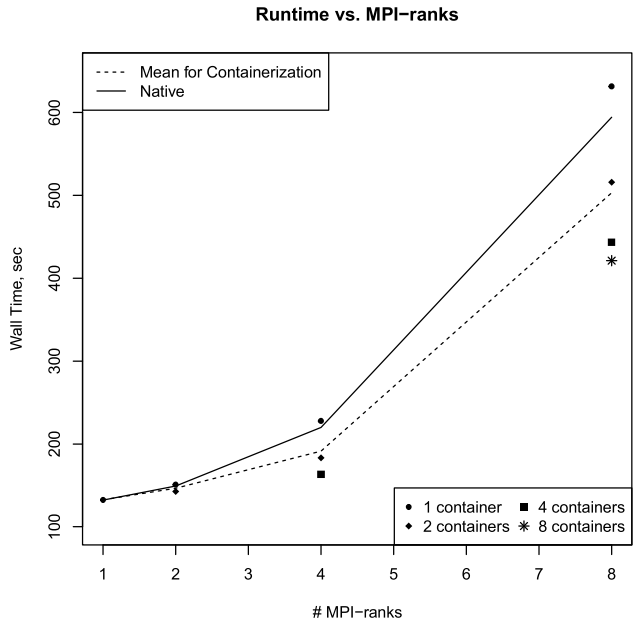


Figure 1: Execution time for scaled input test (1 vnode).

deviation of runtime, system time to wall time ratio and so on, measured for all MPI-processes (i.e. for master rank as well as slave ranks) revealed that in the case of relatively small computational complexity and in the case of parallel multiprocess computation there is no performance degradation of containerised Firefly run versus native Firefly execution. Moreover, choosing the right distribution of containers on nodes it is possible to reduce execution time. We attribute the performance gains to better memory localization and execution of some system calls in user mode by containerised version. Thus we conclude that Firefly can be successfully containerised and take full advantage of simplified software configuration management, resource allocation and capping, as provided by this infrastructure solution.

References

[1] Al-Rayhi N., Salah K., Al-Kork N., et al., 2018, International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 75–80

[2] Li Z., Kihl M., Lu Q., et al., 2017, IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan, 955–962

[3] Nguyen N., Bein D., 2017, IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 1–7

[4] Docker Hub, `https://hub.docker.com/`

[5] Granovksy A., Firefly version 8, `http://classic.chem.msu.su/gran/firefly/index.html`

Microelectronic technology trends and numerical simulation tools

A.A. Moskovsky[1,2]

[1] RSC Group, Moscow, Russia
[2] Chemistry Department, M.V. Lomonosov Moscow State University, Moscow, Russia

While numerical simulation necessity was a primary reason for the modern computer architecture inception, today's computers are mostly used for other applications. Moreover, numerical simulations are insufficient to financially justify development of specialized computational devices in the context of rapid technology development during the last decades (so called Moore's law [1]). Over the last 30 years, we witnessed numerical simulations conducted on devices, primarily targeted at various broad commercial markets. The latest shift is to General Purpose Graphic Processing Units (GP-GPUs). This family of architectures evolved from gaming personal computer (PC) industry to vehicles behind the most recent artificial intelligence (AI) boom. From a point of view of computational chemist, doing some complex modeling, this recent development gives mix of opportunities and challenges. On one hand, modern GPU can elicit, theoretically, hundreds of trillions of operations even for a humble devices, but at the cost. Its needed to re-write old software for CPU-GPU tandem, support some new floating point number formats and abide to more complex memory constraints. The 64-bit precision floating point numbers cannot be considered as supported seamlessly by hardware. Memory hierarchy may now include performant high bandwidth memory (HBM) with low latency, traditional DDR/GDDR and/or even some non-volatile segments.

References

[1] Moore G., 1965, *Electronics Magazine*, 38, 4

ISBN 978-5-00244-253-9

Материалы Международной школы-конференции по современным проблемам химической физики и теоретической химии «CHALLENGES 2024»

Издательство «Перо»
109052, Москва, Нижегородская ул., д. 29-33, стр. 15, ком. 536
Тел.: (495) 973-72-28, 665-34-36
Подписано в печать 01.03.2024. Формат 60×90/16.
Бумага офсетная. Усл. печ. л. 16,25. Тираж 200 экз. Заказ 236.

9 785002 442539