FOUNDRY / ЛИТЕЙНОЕ ПРОИЗВОДСТВО

UDC 621.74
https://doi.org/10.17073/0021-3438-2023-3-38-53

Research article
Научная статья

The efficiency of multithreaded computing in casting simulation software

V.E. Bazhenov[1], **A.V. Koltygin**[1], **A.A. Nikitina**[1], **V.D. Belov**[1], **E.A. Lazarev**[2]

[1] **National University of Science and Technology «MISIS»**
4 Leninskiy Prosp., Moscow, 119049, Russia

[2] **PJSC "UEC-Kuznetsov"**
29 Zavodskoe Shosse, Samara, 443009, Russia

✉ Viacheslav E. Bazhenov (V.E.Bagenov@gmail.com)

Abstract: The utilization of computer simulation software for casting process simulation is becoming essential in the advancement of casting technology in aviation and other high-tech engineering fields. With the increase in the number of computational cores in modern CPUs, the use of multi-threaded computations is becoming increasingly relevant. In this study, the efficiency of multi-threaded computations in modeling casting processes was evaluated using finite element method casting simulation software ProCast and PoligonSoft, which utilize parallel computing architectures with distributed (DMP) and shared (SMP) memory, respectively. Computations were performed on Intel and AMD-based computers, varying the number of computational threads from 4 to 32. The calculation efficiency was evaluated by measuring the calculation speed increase in the filling and solidification of GP25 castings made of ML10 alloy, as well as the complex task of filling and solidification modeling nickel superalloy casing castings with radiation heat transfer simulation. The results indicate that the minimum computation time in ProCast software is observed when using 16 computational threads. This pattern holds true for both computing systems (Intel and AMD processors), and increasing the number of threads beyond this point does not make a practical difference. The performance decrease in this scenario can be attributed to the low-performance energy-efficient cores in systems based on Intel processors or the decrease in core frequency and full loading of physical cores in systems based on AMD processors. Multi-threading the modeling task in PoligonSoft software is less efficient than in ProCast, which is a result of the shared-memory architecture used in PoligonSoft. Despite the significant difference in parallel efficiency, the task of GP25 casting solidification in both PoligonSoft and ProCast is solved in a time close enough to be considered sufficient.

Keywords: casting simulation, multithreaded computing, ProCast, PoligonSoft, SMP, DMP.

Acknowledgments: This research received financial support from the Ministry of Science and Higher Education in the Russian Federation (Agreement No. 075-11-2022-023 from 06 April 2022) under the program "Scientific and technological development of the Russian Federation" according to governmental decree No. 218 dated 9 April 2010.

For citation: Bazhenov V.E., Koltygin A.V., Nikitina A.A., Belov V.D., Lazarev E.A. The efficiency of multithreaded computing in casting simulation software. *Izvestiya. Non-Ferrous Metallurgy.* 2023;29(3):38–53. https://doi.org/10.17073/0021-3438-2023-3-38-53

Эффективность многопоточных вычислений в системах компьютерного моделирования литейных процессов

В.Е. Баженов[1], **А.В. Колтыгин**[1], **А.А. Никитина**[1], **В.Д. Белов**[1], **Е.А. Лазарев**[2]

[1] **Национальный исследовательский технологический университет «МИСИС»**
119049, Россия, г. Москва, Ленинский пр-т, 4, стр. 1

[2] **ПАО «ОДК-Кузнецов»**
443009, Россия, г. Самара, ул. Заводское шоссе, 29

✉ Вячеслав Евгеньевич Баженов (V.E.Bagenov@gmail.com)

Аннотация: Применение систем компьютерного моделирования литейных процессов (СКМ ЛП) становится обязательным при разработке литейной технологии в авиации и других наукоемких областях техники. В связи с увеличением числа расчетных

ядер в современных процессорах актуальным становится осуществление многопоточных вычислений. В работе оценивалась эффективность многопоточных вычислений при моделировании литейных процессов с помощью конечно-элементных СКМ ЛП «ProCast» и «ПолигонСофт», использующих архитектуры параллельных расчетов с распределенной (DMP) и общей (SMP) памятью соответственно. Для вычислений применяли компьютеры на базе платформ от компаний «Intel» и «AMD». Число расчетных потоков варьировали от 4 до 32. Эффективность оценивали по приросту скорости расчета заполнения и затвердевания отливки «ГП25» из сплава МЛ10, а также решения сложной задачи моделирования заполнения и затвердевания корпусных отливок из никелевого жаропрочного сплава с учетом радиационного теплообмена. Показано, что минимальное время расчета в СКМ ЛП «ProCast» наблюдается при использовании 16 вычислительных потоков. Причем это характерно для обеих вычислительных систем (на процессорах «Intel» и «AMD»), и увеличение числа потоков выше этого предела не имеет практического смысла. Снижение производительности в данном случае может быть связано с наличием малопроизводительных энергоэффективных ядер в случае применения системы на процессоре «Intel», а также полной загрузки физических ядер и уменьшением частоты ядер для системы на процессоре от «AMD». Распараллеливание задачи моделирования в СКМ ЛП «ПолигонСофт» менее эффективно, чем в СКМ ЛП «ProCast», вследствие реализации архитектуры с общей памятью. В то же время, несмотря на значительную разницу в эффективности распараллеливания, задача затвердевания отливки «ГП25» в СКМ ЛП «ПолигонСофт» и «ProCast» решается за достаточно близкое время.

Ключевые слова: компьютерное моделирование литейных процессов, многопоточные вычисления, ProCast, ПолигонСофт, SMP, DMP.

Благодарности: Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации в рамках постановления Правительства № 218 по соглашению о предоставлении субсидии № 075-11-2022-023 от 06.04.2022 г. «Создание технологии изготовления уникальных крупногабаритных отливок из жаропрочных сплавов для газотурбинных двигателей, ориентированной на использование отечественного оборудования и организацию современного ресурсоэффективного, компьютероориентированного литейного производства».

Для цитирования: Баженов В.Е., Колтыгин А.В., Никитина А.А., Белов В.Д., Лазарев Е.А. Эффективность многопоточных вычислений в системах компьютерного моделирования литейных процессов. *Известия вузов. Цветная металлургия*. 2023;29(3):38–53.
https://doi.org/10.17073/0021-3438-2023-3-38-53

Introduction

With the advancements in computer technology, simulation has become extensively utilized in various manufacturing fields, including casting. Casting engineers routinely employ numerous casting simulation software (CSS) for this purpose. Nowadays, simulation has become an obligatory step in the process of gating system development, particularly in industries like aerospace and other advanced sectors. The implementation of simulation helps in reducing production engineering costs.

In order to enhance performance, advanced computer-aided engineering (CAE) tools and CSS support multithreaded computations. The initial concept of parallel computing originated from grid computing, which involves a network of interconnected computers forming a “virtual supercomputer”. Presently, cloud computing employing remote supercomputers has become the dominant approach, although multi-core personal computers (PCs) are also widely used [1]. These developments are based on the progress in computer science, particularly the increasing computational speed of multicore processors [2]. Furthermore, GPU computing has emerged as an intriguing advancement, although its current usage in casting simulation is limited [3—5].

Parallel computing is typically categorized into two types: distributed memory processing (DMP) and shared memory processing (SMP). In DMP, each CPU core is allocated a specific memory, whereas in SMP, all cores access the same shared memory (see Fig. 1) [4; 6]. The message passing interface (MPI) is responsible for managing message exchange between multiple cores [7].

The DMP architecture has been proven to be more efficient. For example, Pannala S. et al. [8] conducted an assessment of parallel computations for fluidized bed simulation. They found that the SMP architecture with 32 threads resulted in a 14-fold improvement in performance, whereas the DMP architecture achieved a remarkable 27-fold acceleration under the same conditions, which is close to the theoretical maximum limit of 32.

However, it should be noted that the total increase in computing performance is often is less than the

sum of the performance of individual cores [9]. The SYSWeld (ESI Group) welding and heat treatment simulation software utilizes the DMP architecture, which yields a 5-fold performance improvement with 8 threads [10]. As the number of threads is further increased, the efficiency of computation parallelization diminishes. For 32 threads, the performance gain was only 9-fold, representing a mere quarter of the theoretically possible limit. In the work by Yang W.-H. et al. [11], ANSYS (ANSYS, Inc.) was employed to simulate plastic injection molding with 1 to 4 threads. The performance improvement ranged from 1.5 to 2.1 times for two threads, and 3.1 to 3.4 times for four threads. Corke G. [12] tackled various simulation problems in ANSYS 17 with different thread numbers (up to 28), revealing that the performance gain occurred up to 16 threads. Beyond that point, increasing the number of threads had little to no effect on performance, and in some cases, it even decreased. On average, the performance gain remained below 7-fold for 28 threads. Posey S. et al. [5] reported a 6-fold performance improvement when solving deformable body, CFD, and heat transfer problems with LS-DYNA (ANSYS, Inc.) using 6 threads. The computational performance is not solely determined by CPU speed and the number of threads. Corke G. [12] demonstrated that replacing the hard disk drive (HDD) with a Serial Advanced Technology Attachment (SATA) Solid-state drive (SSD) resulted in a 4-fold performance increase, while a faster NVME (Non-Volatile Memory Host Controller Interface Specification) SSD led to a 7-fold improvement. Expanding the RAM also contributes to the performance gain, although to a lesser extent. Yang W.-H. et al. [11] reported conflicting relationships between the number of computational mesh nodes and computational performance [11].

Parallel computations have also been applied to casting simulation. For example, Trębacz L. et al. [13] conducted simulations of continuous steel casting using a distributed solver and the ProCast CSS (ESI Group). Their findings indicated that the size of the FEM mesh has little impact on the efficiency of parallel computing. The study also revealed that the ProCast solver exhibits low computing costs for thread synchronization. The efficiency of the solver with 8 threads reached only 70 % of the theoretically possible limit compared to single-thread computation. Konopka K. et al. [2] simulated continuous casting using ProCast and employed both distributed computing and a cloud computing platform. Their results demonstrated that 2, 4, and 8 threads reduced the calculation time by 71, 219, and 421 %, respectively, compared to a single thread. This indicates that the performance gain diminishes as the number of threads increases, likely due to increasing costs associated with message exchanges between solver threads.

In summary, parallelism in casting simulation can significantly enhance performance. However, it is important to note that the papers [2] and [13] were published in 2014—2015. Since then, parallel computing has advanced significantly with the emergence of

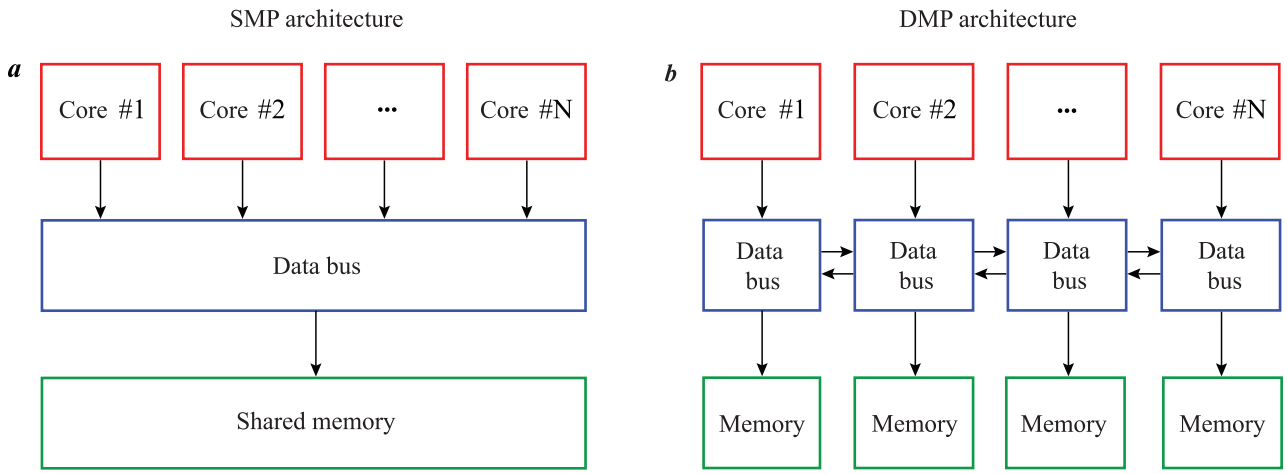

Рис. 1. Архитектуры параллельных вычислений по принципу SMP (***a***) и DMP (***b***)

Fig. 1. Schemes of parallel computing based on the SMP (***a***) and DMP (***b***) architectures

multicore processors capable of running 32 or more threads.

Casting processes are primarily simulated using the finite element method (FEM), finite difference method (FDM), and finite volume method (FVM). FDM (and FVM) solutions are considerably inferior to FEM in terms of accuracy, reliability, computational resource requirements, performance, and postprocessing options [14]. Hence, for casting simulation in this study, we employed FEM.

The objective of this study is to compare the performance of multithreaded casting simulations using ProCast [15; 16] and PoligonSoft [17—19] CSS, which utilize FEM and have DMP and SMP architectures, respectively [13].

Materials and methods

In our study, we conducted simulations of mold filling and solidification using ProCast 2022 (ESI Group, France) and PoligonSoft 2022 (CSoft, Russia). The simulation times were measured for different numbers of threads. ProCast employs FEM to simulate filling and solidification. For solidification analysis, the Fourier heat conduction equation is utilized, taking into account the additional latent heat release. Mold filling is simulated using the Navier-Stokes equation. The filling simulation uses the Newtonian viscosity model and a momentum-driven free surface movement model that considers mass conservation [20]. For further details and equations, please refer to [21—23]. On the other hand, PoligonSoft analyzes solidification using FEM and the Fourier equation [24].

We executed the simulations on two computers equipped with Intel (USA) and AMD (USA) CPUs. The specifications of the computers are provided in Table 1. The Intel system utilized 4, 8, 12, 16, 20, and 24 threads, while the AMD system employed 4, 8, 12, 16, 20, 24, 28, and 32 threads. Throughout the simulations, we monitored the CPU temperature, which remained below the CPU temperature limit, and no throttling (reduction in CPU frequency due to overheating) was necessary. The operating system (OS) and programs running in the background take up some of the computing resources. It is worth noting that the OS and background programs utilize some computing resources. However, in the case of the computers we used, the impact of the OS is assumed to be negligible due to the large number of cores in the processors, excessive RAM allocation for the OS, and, most importantly, the presence of fast SSD M.2 NVME SSDs. To minimize any performance impact, no other applications were running on the computers during the simulations.

In our analysis, we primarily focused on the GP25 cast part, which is manufactured using the ML10 magnesium alloy and cast in resin bonded sand mold with cast iron chill inserts. The GP25 part is a thin-walled, case-shaped part, and its configuration is presented in [25]. To estimate the thermophysical properties of the ML10 alloy, we utilized ProCast thermodynamic properties database for non-equilibrium solidification, employing the Scheil-Gulliver model. Similarly, the properties of the cast iron chill inserts were obtained from the ProCast materials database. The thermophysical properties of the resin bonded sand mold were referenced from Palumbo G. et al. [26], while the casting-mold and casting-chill insert interfacial heat transfer coefficients were obtained from Bazhenov V. et al. [27; 28]. The mold filling time was 40 s., with a melt pouring temperature of 720 °C, and a mold temperature of 20 °C. The mold filling and solidification simulation time were measured separately since the number of FEM mesh nodes varies during filling but remains constant during solidification. This variation may potentially impact the runtime with different thread configurations.

Furthermore, we conducted simulations of filling and solidification for three large-size casing parts (ranging from 705 to 1136 mm in diameter) made of the VZHL14H-VI nickel superalloy, using an investment casting process with ceramic mold. The simulation encompassed various stages, including mold cooling in the furnace before filling, melt filling into the mold, and subsequent cooling. The simulation also accounted for radiation heat transfer. The aim was to assess the efficiency of parallelism in addressing complex, multistage simulation problems. The thermophysical properties of the VZHL14H-VI alloy were calculated using the thermodynamic database of ProCast CSS, while the Scheil-Gulliver model was employed to estimate properties for non-equilibrium solidification. The thermophysical properties of the mold ceramics, steel furnace components, heat insulation, and backing material were derived from the ProCast materials database. The boundary conditions were also retrieved from the database using the investment casting simulation workflow. The initial melt temperature was set at 1450 °C, and the initial mold temperature was 950 °C.

Table 1. **Hardware of computers used for simulation**

Таблица 1. Комплектующие, используемые при сборке компьютеров для моделирования

Name	Intel system	AMD system
Processor	Intel Core I9 12900KF (16 cores, 24 threads)	AMD Ryzen 9 5950X (16 cores, 32 threads)
Motherboard	Gygabyte Gaming X (LGA1700, Z690)	MSI MAG Torpedo (AM4, X570)
RAM	Kingston DIMM DDR5, 5200 MHz, CL40, 64 GB (2 modules)	Kingston DIMM DDR4, 3600 MHz, CL18, 64 GB (2 modules)
Cooling system	NZXT Kraken X73 liquid cooler	ID-Cooling SE-207 XT air cooler
SSD	Samsung 980 Pro, M.2 NVME, 1000 GB	Samsung 970 EVO, M.2 NVME, 500 GB

For mesh generation, the Visual Mesh mesher, a module of ProCast, was employed. The same mesh was used for the PoligonSoft simulations. The GP25 part employed an adaptive mesh with variable element sizes (node-to-node distances), ranging from 3, 4, 5, 6, 8, to 10 mm. These meshes are denoted as L3, L4, L5, L6, L8, and L10, respectively. The finite elements representing the riser and mold were 2 times and 5 times larger, respectively (as indicated in Table 2). The number of tetrahedral finite elements for meshes L3 to L10 ranged from 7.37 to 0.73 mln.

Regarding the FEM meshes representing the casting of large parts made of the VZHL14H-VI nickel superalloy, the element sizes ranged from 3 mm for the part itself to 70 mm for the furnace chamber.

Results and discussion

Figure 2 illutstrates the simulation times for GP25 mold filling using ProCast on Intel and AMD PCs, as well as the corresponding performance gain (expressed as percentage) with respect to the number of threads. These results pertained to finite elements L3—L10 size.

For the Intel PC (Fig. 2, *a*), the simulation time curve exhibits a minimum value across all the meshes, irrespective of the number of elements. The minimum simulation time was achieved when employing 16 threads. However, increasing the thread count to 20 and 24 not only fails to decrease the simulation time but actually leads to an increase. Additionally, the simulation time increases as the number of mesh elements grows.

Figure 2, *b* demonstrates that the performance gain for 8 threads compared to 4 threads ranges from 66 to 77 %, while for 16 threads, it falls between 144 and 172 %. In most cases, meshes with a larger number of smaller elements yield a higher performance gain. The dashed curve represents the maximum theoretical performance gain, assuming that the gain for N threads is equal to the performance of a single thread multiplied by N. Accordingly, the maximum theoretical performance gain for 8 and 16 threads over 4 threads should be 100 and 300 % respectively. However, it is evident that as the threads count increases, the deviation from the maximum possible performance gain becomes more pronounced. Notably, when the number of threads exceeds 16 (20 and 24), a decrease in performance is observed.

While a decrease in performance with increasing thread count has been observed by other researchers studying different simulation software [2; 5; 8; 10—13], the decline seen when the number of threads surpasses 16 appears somewhat peculiar. To shed light on potential causes for this performance degradation, a closer examination of the Intel I9 12900KF processor is warranted. This processor comprises 8 performance cores (designated as “P” for “performance”) and 8 energy-efficient cores (denoted as “E”). The P cores can further divide calculations into 2 threads,

Table 2. **The mesh parameters used for simulation**

Таблица 2. Характеристики сеточных моделей, использованных для моделирования

Part	Initial mesh element size, mm			Number of 3D elements, mln.
	Part	Riser	Mold	
GP25 (L10)	10	20	50	0.73
GP25 (L8)	8	16	40	0.98
GP25 (L6)	6	12	30	1.62
GP25 (L5)	5	10	25	2.35
GP25 (L4)	4	8	20	3.82
GP25 (L3)	3	6	15	7.37
Casing part 1	3–70	3–70	3–70	7.32
Casing part 2	3–70	3–70	3–70	3.52
Casing part 3	3–70	3–70	3–70	4.94

resulting in a total of 16 P and 8 E logical cores. The performance of the P cores significantly surpasses that of the E cores.

Table 3 presents the number of logical cores versus the number of threads, as well as the core clock frequency measured after 5 minutes of the simulation start. For 4 threads, both the performance P cores and energy-efficient cores are utilized. It should be noted that 12 performance logical cores and 7 energy-efficient cores are employed when utilizing 4 threads, which accounts for the majority of available cores. Consequently, the application is distributed among the logical cores, as each physical P core encompasses two logical cores. Only when 4 threads are employed the clock frequencies of individual physical cores increase to 5 GHz and 3.9 GHz for P and E cores, respectively. This factor may account for the high performance gain observed (which is quite close to the theoretically maximum achievable gain) with a small number of threads. However, employing energy-efficient cores proves impractical given their lower performance. After distributing the mesh nodes across the cores, each simulation step is completed more swiftly by the high-performance cores, leading to the performance cores waiting for the energy-efficient cores to finish their part of the calculation.

For 8 or more threads, the clock frequencies of P and E cores are 4.9 GHz and 3.7 GHz, respectively. The utilization of cores varies randomly, but their overall load increases. The case of 16 threads is particularly interesting as only the performance P cores are engaged, representing the maximum performance mode (as shown in Fig. 2, *a* and *b*). Further increasing the thread count results in a slowdown of the simulation due to the simultaneous utilization of energy-efficient E cores and the absence of load sharing among logical cores. For instance, with 20 threads, the E cores bear a 50 % load, with only 4 out of 8 E cores being operational. The remaining E cores remain idle, with a load of 0 %.

In the case of the AMD system (refer to Fig. 2, *c*), a similar trend is observed with the simulation time curve reaching its minimum when the number of threads is 16, mirroring the behavior of the Intel system. Despite the architectural differences between the processors, the optimal performance is achieved when utilizing 16 threads for both systems. However, increasing the number of threads beyond 20 results in a significant increase in simulation time, with the simulation time for 8 threads being shorter than for 20 threads.

Figure 2, *d* illustrates the performance gain for the AMD system. The gain is lower compared to the Intel system, with 44 % to 62 % for 8 threads and 97 % to 116 % for 16 threads. As the number of threads increases, the performance gain deviates further from the dashed

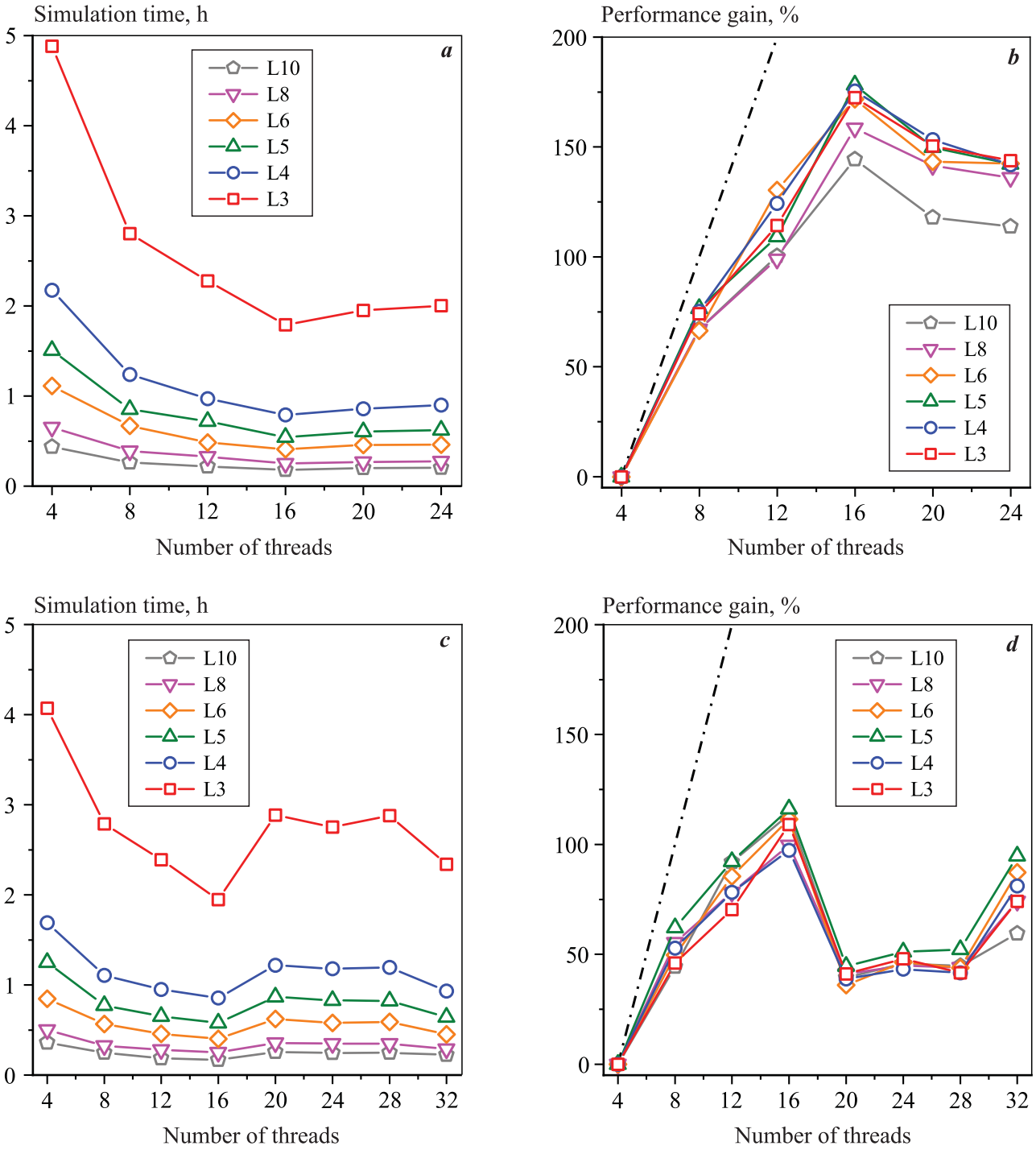

Fig. 2. The calculation time of GP25 casting filling in ProCast (***a***, ***c***) and the calculation speed growth (***b***, ***d***) depending on the number of threads involved in the calculation, when using computers based on the platforms from Intel (***a***, ***b***) and AMD (***c***, ***d***) and L3–L10 simulation meshes (see Table 2)

Рис. 2. Длительность расчета заливки отливки «ГП25» в программе «ProCast» (***a***, ***c***) и прирост скорости расчета (***b***, ***d***) в зависимости от количества потоков, задействованных в расчете при использовании компьютеров на базе платформ «Intel» (***a***, ***b***) и «AMD» (***c***, ***d***) и расчетных сеток L3–L10 (см. табл. 2)

Table 3. **The number of threads involved in the calculation, cores load and frequency**

Таблица 3. Задействованное в расчете количество потоков, загрузка ядер и частота

Number of threads	Intel					AMD				
	P cores		E cores		P/E core clock frequency, GHz	CCX #0		CCX #1		Core clock frequency, GHz
	Number (16 max.)	Load, %	Number (8 max.)	Load, %		Number (16 max.)	Load, %	Number (16 max.)	Load, %	
4	12	13	7	25	4.9–5.0 / 3.7–3.9	16	13	11	13	4.63
8	9	36	5	23	4.9 / 3.7	16	26	14	25	4.58
12	11	49	8	46	4.9 / 3.7	16	51	8	25	4.41
16	16	100	0	0	4.9 / 3.7–3.9	16	53	16	51	4.30
20	16	100	4	50	4.9 / 3.7	16	100	4	26	4.42
24	16	100	8	100	4.9 / 3.7	16	100	8	52	4.31
28	–	–	–	–	–	16	100	12	76	4.19
32	–	–	–	–	–	16	100	16	100	4.08

line representing the maximum theoretical performance gain. Notably, a substantial performance drop occurs when the number of threads exceeds 20, without any apparent correlation between the performance gain and mesh size.

The AMD 5950X processor architecture consists of two CCX (core complex) clusters, each containing 16 logical cores (they are specified as CCX #0 and CCX #1 in Table 3). While sharing some similarities with the Intel processor discussed earlier, the AMD complex cores are identical and lack differentiation between high-performance and energy-efficient cores.

Table 3 provides the number of logical cores and threads for the AMD processor. When using 4 threads, both CCXs clusters are utilized, similar to the Intel processor, with most logical cores being actively engaged in load distribution. Consequently, the physical cores can operate at a relatively high clock frequency of 4.63 GHz. However, as the number of threads increases to 8 and 12, the clock frequency is reduced to 4.58 GHz and 4.41 GHz, respectively, impacting the overall efficiency of parallel computation. The distribution of load between the CCXs becomes uneven for 12 threads, potentially leading to a slowdown in the simulation. At the maximum performance mode of 16 threads, the CCX load is evenly distributed at 50 %, utilizing all available logical cores. In this case, the average clock frequency decreases to 4.3 GHz. When the number of threads reaches 20, an asymmetry in the CCX clusters loads becomes apparent. The first CCX #0 is fully loaded at 100 %, making it impossible to distribute the load across its logical cores, while the second CCX #1 is only 25 % loaded. It is important to note that despite the incomplete loading of CCX #1, there is no load distribution among its logical cores. The average clock frequency even experiences a slight increase compared to the 16 threads simulation.

Further increasing the number of threads results in a decrease in the average CPU clock frequency, reaching 4.08 GHz when utilizing all available threads. Hence, the performance degradation observed with 20 or more threads can be attributed to the uneven load distribution between CCXs and the decreasing of CPU core clock frequency. This leads to a decrease in the efficiency of load distribution among the logical cores as the system approaches the limit of physical cores (which is 16 cores in the CPU). Another possible contributing factor could be the cache memory. When one physical core handles two threads, the cache is shared. With a large number of commands and data, they may not fit into the cache memory, resulting in an increased number of RAM accesses operations.

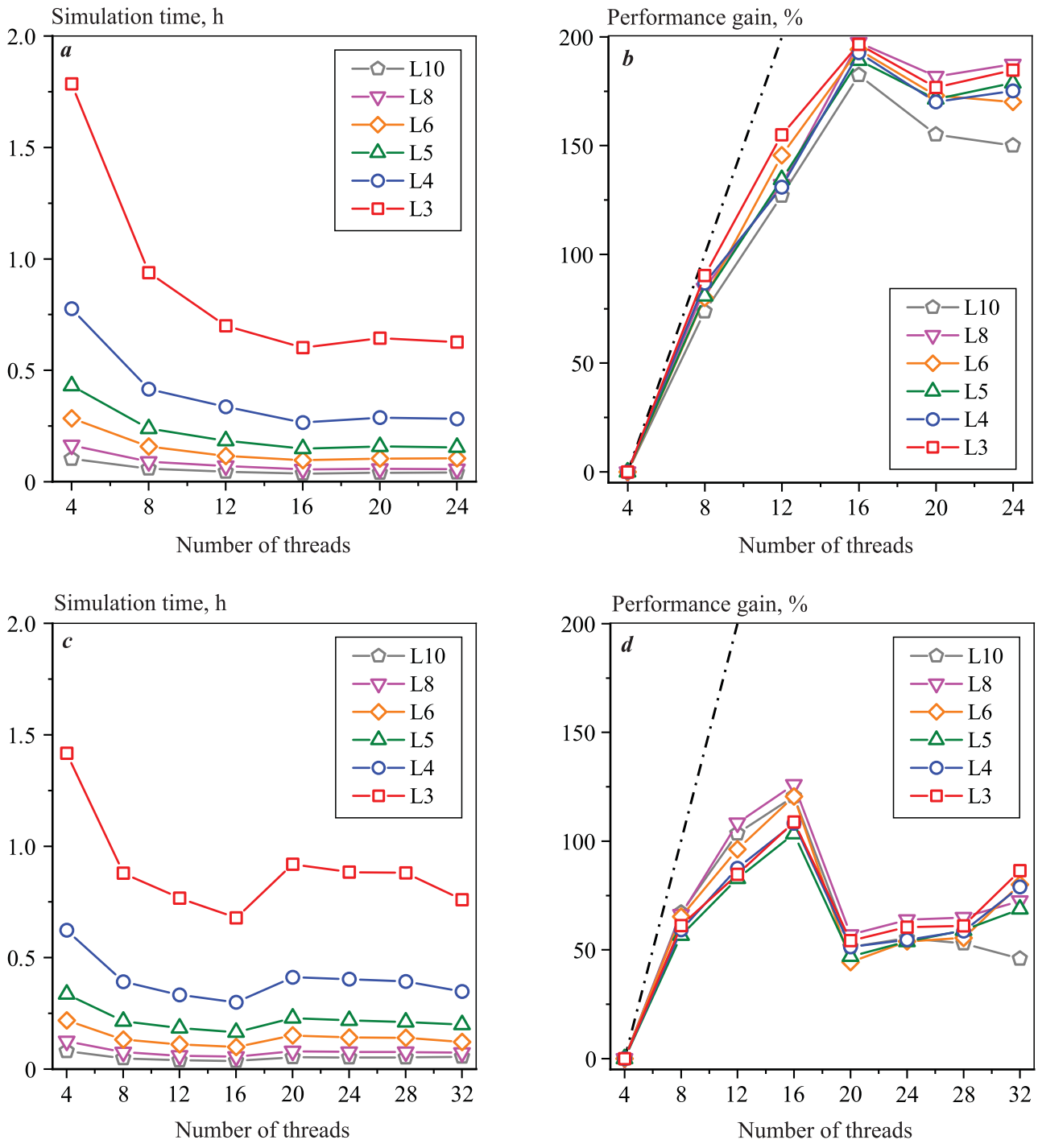

Fig. 3. The calculation time of GP25 casting solidification in ProCast (***a***, ***c***) and the calculation speed growth (***b***, ***d***) depending on the number of threads involved in the calculation, when using computers based on the platforms from Intel (***a***, ***b***) and AMD (***c***, ***d***) and L3–L10 simulation meshes (see Table 2)

Рис. 3. Длительность расчета затвердевания отливки «ГП25» в программе «ProCast» (***a***, ***c***) и прирост скорости расчета (***b***, ***d***) в зависимости от количества потоков, задействованных в расчете при использовании компьютеров на базе платформ «Intel» (***a***, ***b***) и «AMD» (***c***, ***d***) и варьировании размера элементов расчетных сеток L3–L10 (см. табл. 2)

Figure 3 illustrates the simulation time and performance gain for the GP25 part solidification in ProCast on both the Intel and AMD platforms, as well as the relationship with the number of threads. The simulation time and performance gain trends are consistent with those observed in the mold-filling simulation. The only difference is that the solidification simulation is shorter in duration.

Figure 3, *b* presents the performance gain versus the number of threads on the Intel platform. The gain is 73 % to 90 % for 8 threads instead of 4, and 182 % to 197 % for 16 threads instead of 4. Thus, multithreading exhibits slightly higher efficiency in the solidification simulation compared to the mold-filling simulation. This observation holds true for AMD processors as well. One possible explanation for this difference is that during solidification simulation, the distribution of mesh nodes between threads remains constant, whereas during mold-filling simulation, a portion of the CPU time is spent on redistributing nodes between threads.

Figure 4 displays the GP25 part solidification simulation time in PoligonSoft CCS on the Intel platform, along with the performance gain in relation to the number of threads for L3—L10 FEM meshes.

As the number of threads increases, the simulation time decreases and reaches its minimum when using 24 threads. However, the performance gain diminishes with a larger number of threads (Fig. 4, *b*): 23 % to 38 % for 8 threads instead of 4, and 34 % to 65 % for 24 threads. Therefore, increasing the number of threads beyond 8 does not yield significant performance gains. Additionally, as the mesh size decreases and the number of FEM elements increases, there is an almost twofold increase in performance.

Similar to ProCast, analyzing the CPU load is necessary to understand the potential causes of performance degradation and inefficient parallelization of the simulation in PoligonSoft. However, it is challenging to precisely determine how the cores are utilized as the number of threads varies due to PoligonSoft's use of the SMP architecture. The Task Manager

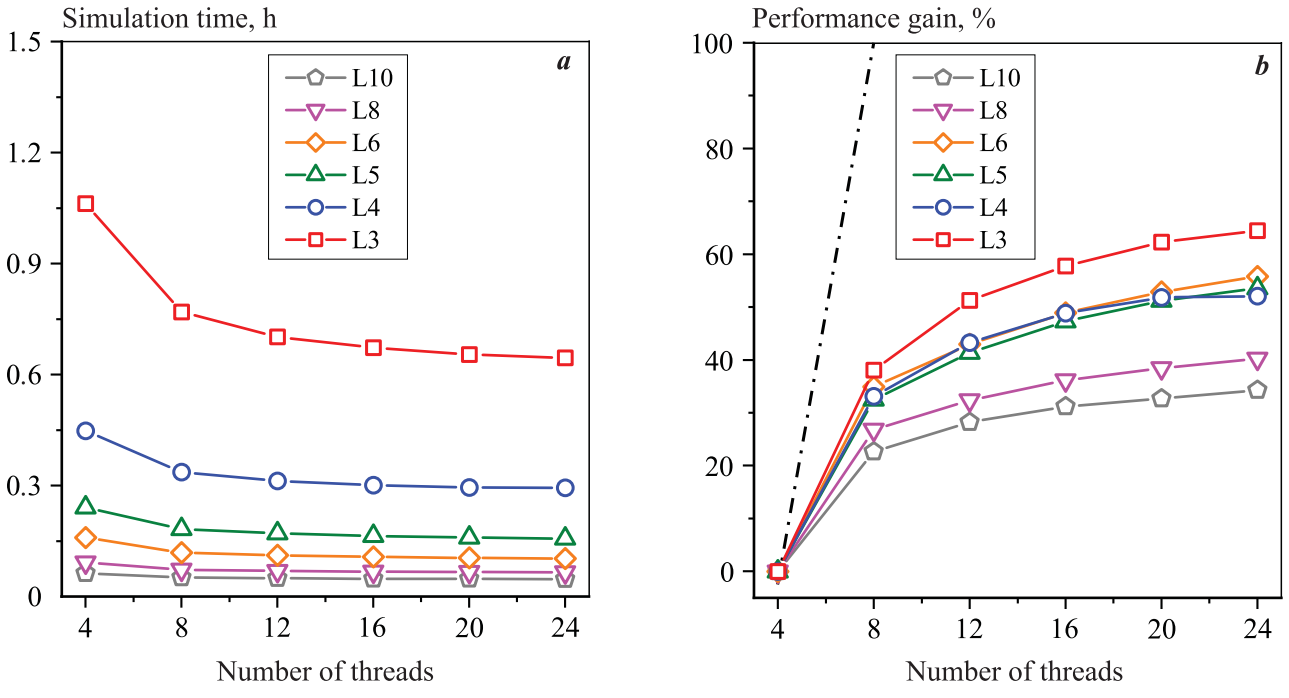

Fig. 4. The calculation time of GP25 casting solidification in PoligonSoft software (***a***) and the calculation speed growth (***b***) depending on the number of threads involved in the calculation, when using computer based on Intel platform and L3—L10 simulation meshes (see Table 2)

Рис. 4. Длительность расчета затвердевания отливки «ГП25» в программе «ПолигонСофт» (***a***) и прирост скорости расчета (***b***) в зависимости от количества потоков, задействованных в расчете при использовании компьютера на базе платформы «Intel» и варьировании размера элементов расчетных сеток L3—L10 (см. табл. 2)

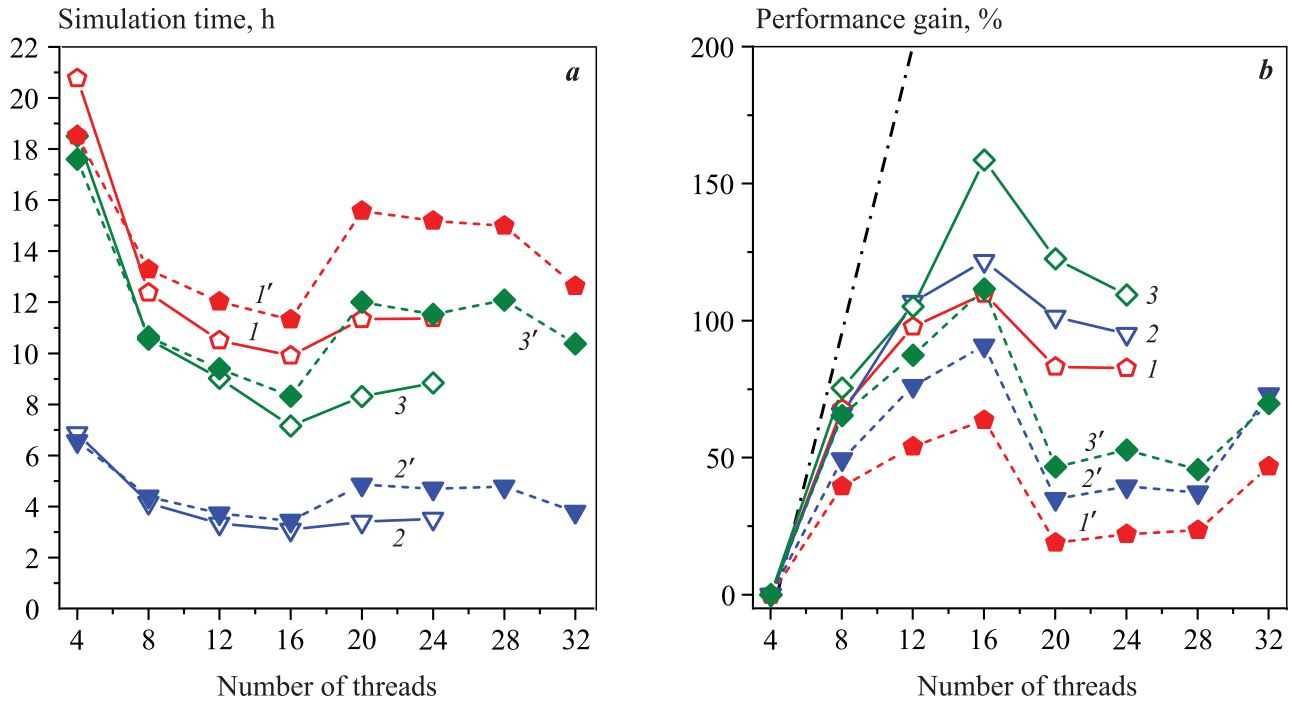

Fig. 5. The calculation time of "body" castings filling and solidification in ProCast (***a***) and the calculation speed growth (***b***) depending on the number of threads involved in the calculation, when using computers based on the platforms from Intel (open points ***1–3***) and AMD (filled points ***1′–3′***)

1, ***1′***, ***2***, ***2′***, ***3***, ***3′*** – numbers of castings "Body"

Рис. 5. Длительность расчета заполнения и затвердевания отливок «Корпус» (см. табл. 2) в программе «ProCast» (***a***) и прирост скорости расчета (***b***) в зависимости от количества потоков, задействованных в расчете при использовании компьютеров на базе платформ «Intel» (пустые точки ***1–3***) и AMD (закрашенные точки ***1′–3′***)

1, ***1′***, ***2***, ***2′***, ***3***, ***3′*** – номера отливок «Корпус»

shows only one process regardless of the actual number of threads. The available information pertains of the load of logical cores from this process, expressed as a percentage for the P and E cores. With 4 threads, only the P cores are loaded, with load varying between 10 % and 22 %. It is worth noting that 22 % represents the peak value reached intermittently, while most of the time, the load remains around 10 %. When the number of threads is increased to 12, both the P cores (20 % to 37 %) and E cores (10 % to 21 %) are utilized. At the maximum number of threads, the load is identical for both P and E cores, and ranging from 20 % to 60 %. The solidification simulation in PoligonSoft significantly underutilizes the CPU due to the less efficient shared memory parallelization (SMP) architecture employed, as opposed to the DMP (Distributed Memory Parallelism) architecture utilized in ProCast, where a specific amount of memory is allocated to each logical core.

Figure 5 illustrates the simulation time for mold filling and solidification of the Casing parts in ProCast, as well as the performance gain in relation to the number of threads for both Intel and AMD computers. This FEM analysis focuses on simulating the cooling of the mold before filling and includes radiation heat transfer. The observed relationships for the GP25 part are consistent with those observed for the Casing part. In both Intel and AMD processors, maximum performance is achieved with 16 threads. With 4 threads, the AMD processor exhibits slightly higher efficiency. However, if the number of threads exceeds 8, the Intel processor becomes more efficient. This difference in performance is likely attributed to the higher clock frequency of the P cores in Intel processor and the use of DDR5 RAM in the Intel computer, as opposed to DDR4 in the AMD computer. The lower performance of the Intel computer with 4 threads is probably due to the predominantly loaded low-performance, energy-efficient E cores (as indicated in Table 3). As the number of threads surpasses 8, the Intel computer utilizes more high-performance cores, while the AMD computer gradually reduces the clock frequency of its

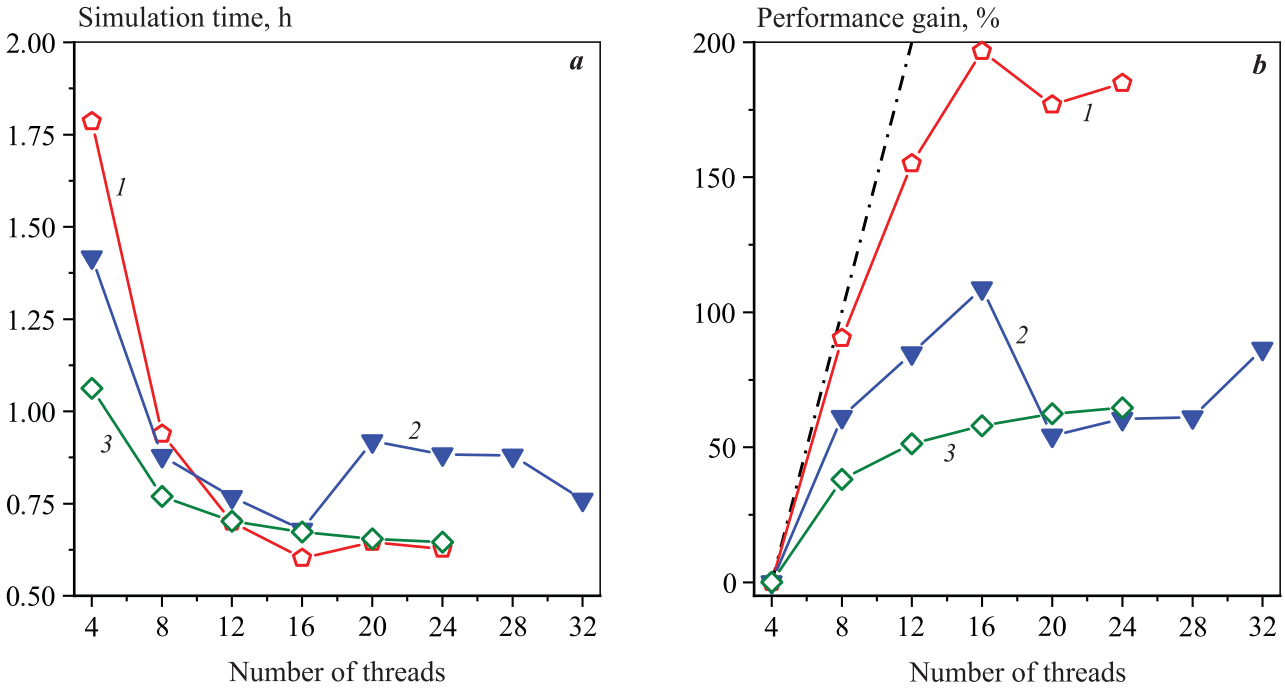

Fig. 6. The calculation time of GP25 casting solidification (***a***) in ProCast (***1***, ***2***) and PoligonSoft (***3***) software and the calculation speed growth (***b***) depending on the number of threads involved in the calculation, when using computers based on the platforms from Intel (open points ***1***, ***3***) and AMD (filled points ***2***) and L3 simulation mesh (see Table 2)

Рис. 6. Длительность расчета затвердевания отливки ГП25 (***a***) и прирост скорости расчета (***b***) в программах «ProCast» (***1***, ***2***) и «ПолигонСофт» (***3***) в зависимости от количества потоков, задействованных в расчете при использовании компьютеров на базе платформ «Intel» (пустые точки ***1***, ***3***) и «AMD» (закрашенные точки ***2***) и расчетной сетки L3 (см. табл. 2)

physical cores. The casting simulation of the GP25 part did not reveal any correlation between the FEM mesh size and the efficiency of parallelism. Similarly, such a relationship was not observed for the Casing parts.

Figure 6 presents the solidification simulation times for the GP25 part in both ProCast and PoligonSoft, as well as the performance gain in relation to the number of threads for Intel and AMD computers. The simulation was conducted using the L3 mesh, which represents the minimum mesh size and the maximum number of FEM elements (as specified in Table 2). Regarding the performance of the ProCast simulation, unlike the casting analysis of the Casing part, the AMD computer demonstrated greater efficiency with 4 and 8 threads, while the Intel computer performed better with a higher number of threads. The potential reasons for this discrepancy are discussed in the earlier section on the Casting part simulation results. It should be noted that the specific number of threads at which one system outperforms the other can vary depending on the specific simulation problem. In general, it is worth noting that the GP25 part solidification simulation in PoligonSoft exhibited faster performance with a small number of threads compared to ProCast. However, when utilizing 16 threads with an Intel processor, ProCast proved to be faster.

It should be noted that the comparison between PoligonSoft and ProCast simulation times is approximate due to potential differences in their FEM algorithms. Therefore, this comparison serves only as a reference and cannot be considered definitive. However, based on the available data, it can be concluded that the solidification simulation times for the same casting parts in ProCast and PoligonSoft are roughly similar.

Despite the diminishing efficiency of parallelism with an increasing number of threads, modern processors equipped with multiple physical and logical cores can still significantly reduce the simulation time for both mold filling and solidification processes. This accelerated simulation enables casting engineers to explore

more casting process options, make better-informed decisions, and ultimately enhance the quality of the castings produced. This is particularly crucial when dealing with large, thin-walled parts, which typically involve a vast number of FEM elements and extended simulation durations.

Conclusions

1. The optimal filling and solidification simulation time for the GP25 part in ProCast is achieved with 16 threads on both the Intel Core I9 12900KF and AMD Ryzen 9 5950X CPUs. By using 16 threads instead of 4, the Intel computer experiences a performance gain of approximately 180 %, while the AMD computer shows a gain of around 110 %.

2. The performance decrease obserbed when using more than 16 threads can be attributed to several factors. In the case of Intel processors, the presence of slower, energy-efficient cores contributes to the decrease in performance. Similarly, AMD processors experience decreasing of CPU frequency, which impacts their overall performance. Additionally, both computers exhibit an uneven load distribution between threads due to the full utilization of physical cores.

3. PoligonSoft's parallel GP25 part casting simulation exhibits significantly lower efficiency compared to ProCast. This is primarily due to PoligonSoft's SMP (shared memory parallelization) architecture, which underutilizes the logical cores available. As a result, the performance gain achieved when using 24 threads instead of 4 does not exceed 65 %.

4. Despite the notable difference in parallelization efficiency, the GP25 part's solidification simulation times in PoligonSoft and ProCast remain relatively close.

5. When conducting a more complex simulation of a large-size VZHL14H-VI nickel superalloy part casting in ceramic molds using ProCast, which includes accounting for mold cooling, melt pouring, and subsequent cooling with radiation heat transfer, the parallelization efficiency remains consistent with that observed in the GP25 part simulation.

References

1. Schwiegelshohn U., Badia R.M., Bubak M., Danelutto M., Dustdar S., Gagliardi F., Geiger A., Hluchy L., Kranzlmüller D., Laure E., Priol T., Reinefeld A., Resch M., Reuter A., Rienhoff O., Rüter T., Sloot P., Talia D., Ullmann K., Yahyapour R., von Voigt G. Perspectives on grid computing. *Future Generation Computer Systems*. 2010;26:1104—1115. https://doi.org/10.1016/j.future.2010.05.010
2. Konopka K., Miłkowska-Piszczek K., Trebacz L., Falkus J. Improving efficiency of ccs numerical simulations through use of parallel processing. *Archives of Metallurgy and Materials*. 2015;60(1):235—238. https://doi.org/10.1515/amm-2015-0037
3. Istomin V.A., Shvarts D.R., Ishkhanov E.A., Tikhomirov M.D. Using the meshfree smoothed-particles method in the simulation of hydrodynamics in the casting simulation software "PoligonSoft". *Liteinoe Proizvodstvo*. 2012;(8):20—22 (In Russ.).
Истомин В.А., Шварц Д.Р., Ишханов Е.А., Тихомиров М.Д. Использование бессеточного метода сглаженных частиц при моделировании гидродинамики в СКМ ЛП «ПолигонСофт». *Литейное производство*. 2012;(8):20—22.
4. Cai Y., Li G., Wang H., Zheng G., Lin S. Development of parallel explicit finite element sheet forming simulation system based on GPU architecture. *Advances in Engineering Software*. 2012;45:370—379. https://doi.org/10.1016/j.advengsoft.2011.10.014
5. Posey S., Kodiyalam S. Performance benefits of NVIDIA GPUs for LS-DYNA®. In: *8th European LS-DYNA Users Conference* (May 2011). Strasbourg, 2011. P. 1—6.
6. Kaplinger B., Wie B., Dearborn D. Efficient parallelization of nonlinear perturbation algorithms for orbit prediction with applications to asteroid deflection and fragmentation. In: *20th AAS/AIAA Space Flight Mechanics Meeting* (15—17 February 2010). San Diego, 2010. P. 1845—1860.
7. Thibault S.E., Holman D., Trapani G., Garcia S. CFD simulation of a quad-rotor UAV with rotors in motion explicitly modeled using an LBM approach with adaptive refinement. In: *55th AIAA Aerospace Sciences Meeting* (9—13 January 2017). Grapevine: American Institute of Aeronautics and Astronautics, 2017. P. 1—16. https://doi.org/10.2514/6.2017-0583
8. Pannala S., D'Azevedo E.F., Syamlal M. O'Brien T. Hybrid (OpenMP and MPI) parallelization of MFIX: A multiphase CFD code for modeling fluidized beds. In: *Proceedings of the ACM Symposium on Applied Computing (SAC)* (9—12 March 2003). Melbourne: Association for Computing Machinery, 2003. P. 199—206. https://doi.org/10.1145/952532.952574

9. Tomsich P., Rauber A., Merkl D. ParSOM: Using parallelism to overcome memory latency in self-organizing neural networks. In: *High Performance Computing and Networking:* Proceedings of the 8th International Conference (8—10 May 2000). Amsterdam: Springer, 2000. P. 136—145. https://doi.org/10.1007/3-540-45492-6_15
10. Bilenko G.A. General capabilities of the software package Welding simulation suite. *Metallurgist.* 2011;55(5—6): 323—327. (In Russ.). https://doi.org/10.1007/s11015-011-9430-6
Биленко Г.А. Общие возможности пакета программ Welding simulation suite. *Металлург.* 2011;(5):28—31.
11. Yang W.-H., Peng A., Liu L., Hsu D.C. Parallel true 3D CAE with hybrid meshing flexibility for injection molding. In: *Annual Conference of the Society of Plastics Engineers ANTEC 2005* (1—5 May 2005). Boston: Curran Associates Inc., 2005. Vol. 1. P. 56—60.
12. Corke G. Workstations for simulation (FEA). URL: https://develop3d.com/features/workstations-for-simulation-fea-ansys-mechanical-17-0 (accessed: 17.01.2023).
13. Trębacz L., Miłkowska-Piszczek K., Konopka K., Falkus J. Numerical simulation of the continuous casting of steel on a grid platform. In: *eScience on distributed computing infrastructure. Lecture notes in computer science* (Eds. M. Bubak, J. Kitowski, K. Wiatr). Heidelberg: Springer, 2014. Vol. 8500. P. 407—418. https://doi.org/10.1007/978-3-319-10894-0_29
14. Tikhomirov M.D., Komarov I.A. Principles of simulation of casting processes. Which is better, the finite element method or the finite difference method? *Liteinoe Proizvodstvo.* 2002;(5):22—28. (In Russ.).
Тихомиров М.Д., Комаров И.А. Основы моделирования литейных процессов. Что лучше — метод конечных элементов или метод конечных разностей? *Литейное производство.* 2002;(5):22—28.
15. Tarasevich N.I., Korniets I.V., Tarasevich I.N., Dudchenko A.V. Comparative analysis of computer simulation software for metallurgical and foundry processes. *Metall i Lit'e Ukrainy.* 2010;(5):20—25. (In Russ.).
Тарасевич Н.И., Корниец И.В., Тарасевич И.Н., Дудченко А.В. Сравнительный анализ систем компьютерного моделирования металлургических и литейных процессов. *Металл и литье Украины.* 2010;(5):20—25.
16. Simulate the complete casting process to reach zero defects with a single tool. URL: https://www.esi-group.com/products/casting (accessed: 6.02.2023).
17. Turishchev V. Simulation of foundry processes: What to choose? *CADmaster.* 2005;(2):33—35. (In Russ.).
Турищев В. Моделирование литейных процессов: Что выбрать? *CADmaster.* 2005;(2):33—35.
18. Casting processes computer simulation system (CSS CP) «PoligonSoft». (In Russ.). URL: https://poligonsoft.ru (accessed: 6.02.2023).
Система компьютерного моделирования литейных процессов (СКМ ЛП) «ПолигонСофт». URL: https://poligonsoft.ru (дата обращения: 6.02.2023).
19. Monastyrskii A., Tikhomirov M. The casting simulation software "PoligonSoft" 13.X. Overview, results, plans. *CADmaster.* 2013;(2):44—48. (In Russ.).
Монастырский А., Тихомиров М. СКМ ЛП «ПолигонСофт» 13.X. Обзор, итоги, планы. *CADmaster.* 2013;(2):44—48.
20. ESI Group, ProCAST 2010.0 User's Manual (ESI Group, 2010). URL: https://myesi.esi-group.com/system/files/documentation/ProCAST/2010/ProCAST_20100_UM.pdf (accessed: 12.09.2022).
21. Yang L., Chai L.H., Liang Y.F., Zhang Y.W., Bao C.L., Liu S.B., Lin J.P. Numerical simulation and experimental verification of gravity and centrifugal investment casting low pressure turbine blades for high Nb—TiAl alloy. *Intermetallics.* 2015;66:149—155. https://doi.org/10.1016/j.intermet.2015.07.006
22. Lu S., Xiao F., Guo Z., Wang L., Li H., Liao B. Numerical simulation of multilayered multiple metal cast rolls in compound casting process. *Applied Thermal Engineering.* 2016;93:518—528. https://doi.org/10.1016/j.applthermaleng.2015.09.114
23. Dantzig J.A., Rappaz M. Solidification. Lausanne: EPFL Press, 2009. 621 p.
24. Tikhomirov M.D. Main aspects of solving the heat transfer problem in the simulation of foundry processes. *Liteinoe Proizvodstvo.* 1998;(4):30—34. (In Russ.).
Тихомиров М.Д. Основные аспекты решения тепловой задачи при моделировании литейных процессов. *Литейное производство.* 1998;(4):30—34.
25. Koltygin A.V., Bazhenov V.E., Tseloval'nik Yu.V., Belov V.D., Yudin V.A. Results of computer of shrinkage microporosity simulation in ML10 alloy body casting. *Liteinoe Proizvodstvo.* 2020;(8):23—27. (In Russ.).

Колтыгин А.В., Баженов В.Е., Целовальник Ю.В., Белов В.Д., Юдин В.А. Результаты компьютерного моделирования усадочной микропористости в корпусной отливке из сплава МЛ10. *Литейное производство*. 2020;(8):23—27.

26. Palumbo G., Piglionico V., Piccininni A., Guglielmi P., Sorgente D., Tricarico L. Determination of interfacial heat transfer coefficients in a sand mould casting process using an optimised inverse analysis. *Applied Thermal Engineering*. 2015; 78:682—694. https://doi.org/10.1016/j.applthermaleng.2014.11.046
27. Bazhenov V.E., Petrova A.V., Koltygin A.V., Tseloval'nik Yu.V. Determination of the interfacial heat transfer coefficient between AZ91 alloy casting and resin bonded sand mold. *Tsvetnye Metally*. 2017;(8):89—96. (In Russ.). https://doi.org/10.17580/tsm.2017.08.14
Баженов В.Е., Петрова А.В., Колтыгин А.В., Целовальник Ю.В. Определение коэффициента теплопередачи между отливкой из сплава МЛ5 (AZ91) и формой из холоднотвердеющей смеси. *Цветные металлы*. 2017;(8):89—96.
28. Bazhenov V.E., Tselovalnik Yu.V., Koltygin A.V., Belov V.D. Investigation of the interfacial heat transfer coefficient at the metal—mold interface during casting of an A356 aluminum alloy and AZ81 magnesium alloy into steel and graphite molds. *International Journal of Metalcasting*. 2021;15(2):625—637. https://doi.org/10.1007/s40962-020-00495-2

Information about the authors

Viacheslav E. Bazhenov – Cand. Sci. (Eng.), Assistant Prof., Department of Foundry Technologies and Material Art Working (FT&MAW), National University of Science and Technology "MISIS" (NUST MISIS).
https://orcid.org/0000-0003-3214-1935
E-mail: V.E.Bagenov@gmail.com

Andrey V. Koltygin – Cand. Sci. (Eng.), Assistant Prof., Department of FT&MAW, NUST MISIS.
https://orcid.org/0000-0002-8376-0480
E-mail: misistlp@mail.com

Anna A. Nikitina – Educat. Master, Department of FT&MAW, NUST MISIS.
https://orcid.org/0000-0002-5399-0330
E-mail: nikitina.misis@gmail.com

Vladimir D. Belov – Dr. Sci. (Eng.), Head of the Department of FT&MAW, NUST MISIS.
https://orcid.org/0000-0003-3607-8144
E-mail: vdbelov@mail.ru

Evgenii A. Lazarev – Head Metallurgist of the PJSC "UEC-Kuznetsov".
E-mail: ea.lazarev@uec-kuznetsov.ru

Информация об авторах

Вячеслав Евгеньевич Баженов – к.т.н., доцент кафедры литейных технологий и художественной обработки материалов (ЛТиХОМ) Национального исследовательского технологического университета «МИСИС» (НИТУ МИСИС).
https://orcid.org/0000-0003-3214-1935
E-mail: V.E.Bagenov@gmail.com

Андрей Вадимович Колтыгин – к.т.н., доцент кафедры ЛТиХОМ, НИТУ МИСИС.
https://orcid.org/0000-0002-8376-0480
E-mail: misistlp@mail.ru

Анна Андреевна Никитина – учебный мастер кафедры ЛТиХОМ, НИТУ МИСИС.
https://orcid.org/0000-0002-5399-0330
E-mail: nikitina.misis@gmail.com

Владимир Дмитриевич Белов – д.т.н., заведующий кафедрой ЛТиХОМ, НИТУ МИСИС.
https://orcid.org/0000-0003-3607-8144
E-mail: vdbelov@mail.ru

Евгений Алексеевич Лазарев – главный металлург ПАО «ОДК-Кузнецов».
E-mail: ea.lazarev@uec-kuznetsov.ru

Contribution of the authors

V.E. Bazhenov – conceptualization, realization of simulation, analysis of the simulation results, writing of the manuscript.

A.V. Koltygin – scientific guidance, review and editing of the manuscript.

A.A. Nikitina – realization of simulation, analysis of the simulation results.

V.D. Belov – supervision, review and editing of the manuscript.

E.A. Lazarev – formulation of the aims and objectives of the study, provision of resources.

Вклад авторов

В.Е. Баженов – формирование основной концепции, проведение расчетов, обработка результатов расчетов, написание текста статьи.

А.В. Колтыгин – научное руководство, редактирование текста статьи.

А.А. Никитина – проведение расчетов, обработка результатов расчетов.

В.Д. Белов – общее руководство, редактирование текста статьи.

Е.А. Лазарев – формулировка цели и задачи исследования, обеспечение ресурсами.

The article was submitted 09.02.2023, revised 22.03.2023, accepted for publication 24.03.2023

Статья поступила в редакцию 09.02.2023, доработана 22.03.2023, подписана в печать 24.03.2023