

МЕТОДЫ И ПРОГРАММНАЯ АРХИТЕКТУРА ВЫЯВЛЕНИЯ НЕЭФФЕКТИВНО ИСПОЛЬЗУЕМЫХ РЕСУРСОВ В СУПЕРКОМПЬЮТЕРНЫХ СРЕДАХ

С.С. Филиппов^{*}, Д.В. Беляков², М.И. Зуев², А.А. Кокорев², Д.И. Пряхина^{1,2}

¹ Государственный университет «Дубна», г. Дубна, Россия

² Объединенный институт ядерных исследований, г. Дубна, Россия

e-mail: fss.25@uni-dubna.ru

Избыточное резервирование вычислительных ресурсов при выполнении разнородных задач приводит к снижению эффективности использования современных центров обработки данных (ЦОД). Особенно ярко выражена проблема неэффективно используемых ресурсов в суперкомпьютерных средах при выполнении параллельных задач [1]. Системы планирования задач (например, Slurm) выделяют ресурсы статично на весь период выполнения задачи. Они не отслеживают динамическое внутризадачное высвобождение оперативной памяти и процессорного времени при досрочном физическом завершении отдельных процессов [2]. Это подтверждает актуальность исследования, которое направлено на разработку методов и программной архитектуры аналитического модуля для автоматизированного количественного выявления скрытых потерь ресурсов в суперкомпьютерных средах.

Для исследования предметной области применялся системный анализ, а проектирование модуля и обработка телеметрии осуществлялись методами архитектурного моделирования и численного интегрирования. Объектом исследования является процесс распределения и использования ресурсов в высокопроизводительных вычислительных кластерах (НРС). Предметом исследования выступают динамические характеристики использования процессорного времени и оперативной памяти при выполнении параллельных MPI-задач. Дисбаланс вычислений между процессами (рангами) таких задач часто приводит к простаиванию выделенных узлов.

Для сбора системной телеметрии выбран программный комплекс Performance Co-Pilot (PCP) [3]. Вместо попроцессного (per-process) мониторинга используется технология агрегированного сбора метрик на уровне контрольных групп (cgroups) операционной системы (ОС). Данный подход обеспечивает изолированную гранулярность телеметрии в строгой привязке к задачам планировщика при минимальных накладных расходах на ядро ОС [4].

Для проведения исследований спроектирована архитектура подсистемы мониторинга. На базе гетерогенной вычислительной платформы HybriLIT [5] развернут экспериментальный стенд, эмулирующий работу НРС-кластера. Стенд, состоящий из семи виртуальных машин на базе ОС AlmaLinux, включает управляющий и вычислительные узлы, объединенные распределенной файловой системой Lustre.

Для обработки телеметрии предложена математическая модель на основе вычисления потерь процессорного времени и памяти при выполнении параллельных программ (формулы 1 и 2):

$$W_{CPU} = \int_0^T (R_{alloc} - r(t)) \cdot (1 - M_{wait}(t)) dt$$

$$W_{MEM} = (M_{alloc} - MaxRSS) \cdot T.$$

Потери процессорного времени W_{CPU} (в ядро-часах) вычисляются за время выполнения задачи T как интеграл разности между выделенной квотой R_{alloc} и фактическим потреблением $r(t)$ в каждый текущий момент времени t . Бинарная маска M_{wait} исключает из расчета потерь периоды простоя, вызванные системными задержками (такими как ожидание ответа от файловой системы), предотвращая ложное завышения количественной оценки неэффективности. Абсолютные потери памяти W_{MEM} (в гигабайт-часах) оцениваются по разнице лимита M_{alloc} и пикового значения $MaxRSS$, поскольку память в НРС является статично блокируемым ресурсом [6].

На основе полученных математических оценок сформирован концептуальный алгоритм выявления паттернов параллельных программ (см. рис. 1). Как показано на схеме, алгоритм агрегирует телеметрию при первом запуске задачи и рассчитывает потери ресурсов. При повторном запуске той же задачи алгоритм генерирует адаптированные параметры. На стенде успешно настроена агрегация телеметрии параллельных задач, что подтверждает техническую реализуемость сбора данных для предложенной модели.

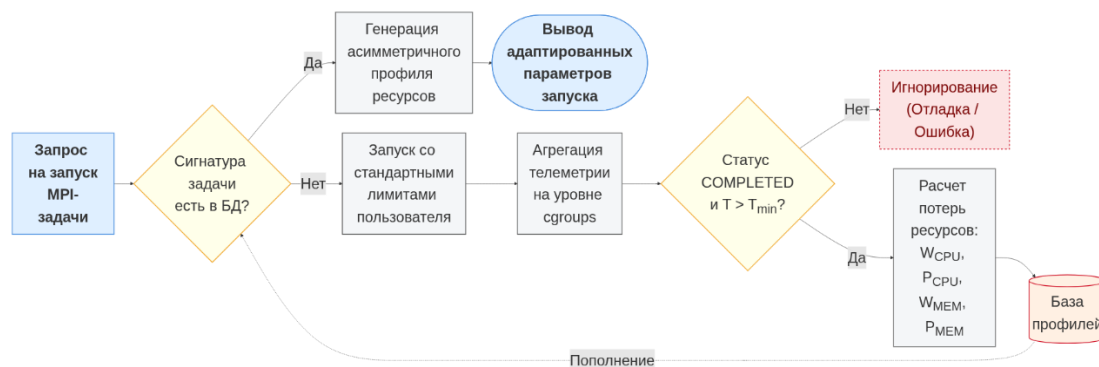


Рис. 1. Блок-схема алгоритма работы аналитического модуля

Для каждого классифицируемого паттерна модуль определяет специфичный путь адаптации. Так, постоянная асимметрия нагрузки компенсируется переходом к комбинированным профилям запуска (Slurm Heterogeneous Jobs), где задаются разные лимиты для разных узлов, а «каскадное завершение» — уменьшением числа выделяемых узлов (Scale Down).

Таким образом, в работе предложена методика и спроектирована архитектура программного модуля, способного выявлять неэффективно используемые ресурсы НРС-кластеров. Отказ от попроцессного сбора в пользу агрегированной телеметрии на уровне задачи целиком и использование строгой математической модели создает основу для генерации скорректированных рекомендаций по резервированию ресурсов без модификации исходного кода пользователя. Результаты могут быть применены для повышения производительности суперкомпьютерных центров за счет уплотнения расписания планировщика. Дальнейшие исследования предполагают программную реализацию классификатора, проведение эмпирической валидации алгоритма на физическом оборудовании, а также интеграцию разработанного модуля в программный комплекс имитационного моделирования цифровых двойников ЦОД [7].

Литература

1. Nikitenko D.A. et al. Supercomputer application integral characteristics analysis for the whole queued job collection of large-scale HPC systems // Параллельные вычислительные технологии (ПаВТ'2016). 2016. С. 20–30.
2. Jette M., Grondona M. SLURM: Simple Linux Utility for Resource Management // ClusterWorld Conference and Expo. — Lawrence Livermore National Laboratory, 2003. [Электронный ресурс]. URL: https://slurm.schedmd.com/slurm_design.pdf (дата обращения: 02.04.2026).
3. Quentin L. Scalability Evaluation of Prometheus for HPC Monitoring [Электронный ресурс]. URL: https://hps.vi4io.org/_media/teaching/autumn_term_2023/stud/hpcsa_lars_quentin.pdf (дата обращения: 02.04.2026).
4. Performance Co-Pilot (PCP) [Электронный ресурс]. URL: <https://pcp.io/> (дата обращения: 02.04.2026)
5. Anikina A., et al. Structure and Features of the Software and Information Environment of the HybriLIT Heterogeneous Platform // Vishnevsky V.M., Samouylov K.E., Kozyrev D.V. (eds.) Distributed Computer and Communication Networks. DCCN 2024. Lecture Notes in Computer Science, vol. 15460. Cham: Springer, 2025. С. 444–457. DOI: 10.1007/978-3-031-80853-1_33.
6. Huber D. et al. Design Principles of Dynamic Resource Management for High-Performance Parallel Programming Models [Электронный ресурс]. URL: <http://arxiv.org/pdf/2403.17107.pdf> (дата обращения: 03.04.2026).
7. Пряхина Д.И., Кореньков В.В., Трофимов В.В. Метод построения цифровых двойников для решения задач эффективного управления и развития распределенных центров сбора, хранения и обработки данных // Современные информационные технологии и ИТ-образование. 2023. Т. 19, № 2. С. 272–281. DOI: 10.25559/SITITO.019.202302.272-281. ISSN 2411-1473.