DOI: 10.15514/ISPRAS-2022-34(2)-13

Реализация функций управления задачами и ресурсами высокопроизводительной вычислительной системы в «СПО Супер-ЭВМ»

А.О. Игнатьев, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@mail.ru>
А.А. Калинин, ORCID: 0000-0001-6152-579X <feycheg@yandex.ru>
С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <sumo@rambler.ru>
Всероссийский НИИ технической физики имени академика Е.И. Забабахина,
456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13

Аннотация. В данной работе приводится общее описание программного обеспечения Slurm-ВНИИТФ, разработки ФГУП РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина, включая его архитектуру и возможности по управлению ресурсами и планированию прохождения задач на высокопроизводительных вычислительных системах, предназначенных для решения задач численного моделирования (ВВС). Проведенные в ходе многолетних работ, связанных с эксплуатацией ВВС, исследования, показывают, что базовых возможностей программного обеспечения Slurm (Simple linux utility for resource management) явно недостаточно для эффективного использования вычислительных ресурсов в крупных вычислительных центрах, поэтому авторами данной публикации предлагается усовершенствованная политика управления задачами и ресурсами, описываются модули расширения (плагины) к Slurm, разработанные в ФГУП РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина и реализующие эту политику.

Ключевые слова: высокопроизводительная вычислительная система; кластер; подсистема управления задачами и ресурсами; Slurm; Slurm-ВНИИТФ; высокопроизводительные вычисления; моделирование.

Для цитирования: Игнатьев А.О., Калинин А.А, Мокшин С.Ю. Реализация функций управления задачами и ресурсами высокопроизводительной вычислительной системы в «СПО Супер-ЭВМ». Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 159-178. DOI: 10.15514/ISPRAS–2022–34(2)–13

Task and resources management function in HPC operation system «SPO Super-EVM»

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@mail.ru>
A.A. Kalinin, ORCID: 0000-0001-6152-579X <feycheg@yandex.ru>
S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <sumo@rambler.ru>
E. I. Zababakhin All-Russian Scientific Research Institute of Technical Physics,
13, Vasilieva street, Chelyabinsk region, Snezhinsk, 456770, Russia

Abstract. The Slurm-VNIITF software developed by Federal State Unitary Enterprise “Russian Federal Nuclear Center - Zababakhin All-Russian Research Institute of Technical Physics”, it’s architecture, resource management capabilities and task management for numerical simulation HPC systems described in this paper. During many years usage of the HPC systems researches show that the basic features of the Slurm (Simple linux utility for resource management) software are clearly insufficient for the effective use of computing resources in HPC centers. Therefore, the authors of this paper propose an improved task and resource management policy. Slurm extension modules (plugins) for implementing this policy also described in this paper.

Keywords: high-performance computing system; cluster; computer simulation; resource management; operation system; HPC modeling; HPC

For citation: Ignatyev A.O., Kalinin A.A., Mokshin S.Yu. Task and resources management function in HPC operation system «SPO Super-EVM». Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 159-178 (in Russian). DOI: 10.15514/ISPRAS–2022–34(2)–13

1.Введение

Многолетний опыт эксплуатации высокопроизводительных вычислительных систем, предназначенных для решения задач численного моделирования (далее по тексту – ВВС) [1] в ФГУП РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина (далее по тексту РФЯЦ-ВНИИТФ) показал, что возможность эффективного использования вычислительных ресурсов ВВС безусловно связана с реализуемыми подходами по функциональной организации расчетов задач, управления их очередью. В ВВС эти функции возлагаются на специализированную подсистему управления задачами и ресурсами, которая предназначена для запуска и выполнения задач пользователей (прикладных программ) в пакетном и интерактивном режиме на вычислительном поле ВВС. В 2015 году в РФЯЦ-ВНИИТФ была разработана предназначенная для использования на ВВС операционная система «СПО Супер-ЭВМ» [2].

Одним из компонентов «СПО Супер-ЭВМ» является программная часть этой подсистемы управления задачами и ресурсами, основанная на программном обеспечении (далее по тексту ПО) Slurm (Simple linux utility for resource management) [3], разработанном в Ливерморской национальной лаборатории им. Лоуренса (Lawrence Livermore National Laboratory) и функционально дополненным в РФЯЦ-ВНИИТФ. Проведенные в ходе многолетних работ, связанных с эксплуатацией ВВС, исследования, показывают, что базовых возможностей ПО Slurm явно недостаточно для эффективного использования вычислительных ресурсов в крупных вычислительных центрах, поэтому авторами данной публикации предлагается усовершенствованная политика управления задачами и ресурсами, описываются модули расширения (плагины) к Slurm, разработанные в РФЯЦ-ВНИИТФ и реализующие эту политику.

В тексте документа используются следующие понятия:

- «базовое ПО Slurm» и «базовые возможности» по отношению к свободно распространяемой версии ПО Slurm;
- «расширенные возможности» применительно к функциям, реализованным в плагинах, разработанных в РФЯЦ-ВНИИТФ (ПО Slurm-ВНИИТФ);
- «задача» (в терминологии Slurm job) – единица планирования выделения ресурсов для проведения определенной вычислительной работы;
- «задание» – описание задачи, подготовленное пользователем для ввода ее в Slurm, содержащее требуемые для задачи ресурсы и выполняемые действия (сценарий выполнения задачи);
- процесс (в терминологии Slurm task) – часть параллельной задачи, запускаемыми компонентами ПО Slurm на вычислительных узлах ВВС.

2. Описание структуры базовой версии SLURM

Для четкого понимания возможностей и пределов модификации базового ПО Slurm приведем описание его структуры, оценим реализованные функции по управлению ресурсами, управлению планированием запуска и запуском задач.

ПО Slurm является высокомасштабируемой программной системой с открытым исходным кодом, предназначенной для управления ресурсами вычислительного комплекса, планирования и запуска вычислительных задач. ПО Slurm выполняет три ключевые функции:

- определяет и захватывает ресурсы (вычислительные узлы, оперативная память,

процессоры, ядра) для пользователей в необходимом количестве на определённое время;
- предоставляет средства для запуска и мониторинга задач на выделенных узлах;
- организует очередь задач, выполняет планирование запуска задач согласно настроенным правилам, предотвращает конфликты при захвате ресурсов.

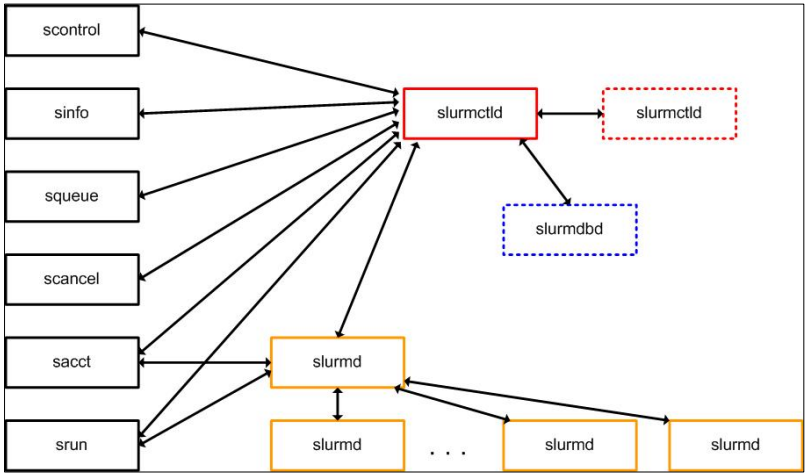

Рис. 1. Состав ПО Slurm и его компонентов
Fig. 1. Slurm structure

ПО Slurm состоит из следующих основных компонентов (см. рис. 1).

- Центральный демон slurmctld (контроллер) – управляет работой всей подсистемы и обрабатывает запросы пользовательских команд. Как правило, для ВВС в целях отказоустойчивости дополнительно может функционировать резервный контроллер, запущенный на другом физическом или виртуальном сервере. Резервный контроллер получает управление в случае отказа основного сервера. Контроллер выполняет функции менеджера ресурсов и планировщика задач.
- Клиентские демоны slurmd, работающие на каждом вычислительном узле. Все клиентские демоны взаимодействуют друг с другом и другими компонентами через сеть, образуя отказоустойчивую иерархическую структуру.
- Утилиты (команды), предназначенные для взаимодействия пользователя (или администратора) с Slurm;
- Необязательный демон slurmdbd, который может использоваться для сохранения статистической и учётной информации по задачам, пользователям и ресурсам в базу данных. Один демон slurmdbd может использоваться для нескольких кластеров и хранить информацию в специально предназначенной для этого базе данных.

Логические объекты, управляемые ПО Slurm, включают следующие сущности:

- узлы (nodes) – вычислительные ресурсы;
- разделы (partitions) – узлы, логически объединённые в одно множество;
- резервации (reservation) – выделенные для специальных целей узлы, логически объединённые в одно множество;
- задачи (задания, jobs) – заданная пользователем вычислительная работа, под которую требуется выделить ресурсы ВВС на определенное время;
- шаги задания (job steps) – множества процессов (экземпляров программы) внутри задачи, обрабатываемые один за другим.

Разделы и резервации можно рассматривать как целевой ресурс планирования очереди задач, каждая из которых имеет набор ограничений, таких как: число выделяемых для задачи узлов, разрешённые пользователи и т.д. Узлы внутри раздела выделяются для задач до тех пор, пока достаточно ресурсов (узлов, процессоров, памяти и т.д.). После того, как под задачу выделен набор узлов, пользователь может инициировать шаги задания (запускать параллельные программы) в любой конфигурации на выделенных ресурсах. Например, один шаг может использовать сразу все узлы, выделенные для задачи, или несколько пунктов могут независимо использовать только часть узлов.

При постановке задачи пользователя в очередь на исполнение ПО Slurm запоминает идентификатор пользователя, поставившего задачу, и связанный с ним идентификатор группы. В дальнейшем, после выделения вычислительных ресурсов, ПО Slurm осуществляет запуск процессов выполняемых в задаче программ и назначает им ранее сохраненные идентификаторы пользователя и группы. Таким образом, процессы на выделенных задаче узлах имеют те же права доступа к данным, что и пользователь, запустивший задачу.

3. Возможности базовой версии SLURM по управлению ресурсами

Большая часть задаваемых при постановке задачи в очередь параметров описывает требуемые для задачи ресурсы. В первую очередь это имя требуемого раздела или резервации. Если они не указаны, используется раздел, заданный администратором как основной. Под разделом подразумевается множество узлов, объединенных общими свойствами (архитектура, характеристики, назначение). Одна задача не может занимать узлы из разных разделов, хотя одни и те же узлы могут входить в разные разделы. Разделы создаются администратором при конфигурировании ПО Slurm на ВВС.

Резервацией также является множество узлов, но, в отличие от разделов, резервации создаются для выполнения специальных действий, таких как настройка узлов, или выполнение задач в заданный промежуток времени. Резервации также создаются администратором, но, по мере необходимости, могут быть созданы или удалены в командном режиме без внесения изменений в конфигурационные файлы ПО Slurm. В отличие от разделов, резервации снабжаются списком доступа, т.е. не все пользователи по умолчанию допускаются к конкретной резервации.

Следующая группа параметров включает в себя информацию о количестве требуемых для задачи вычислительных узлов, а также о максимальном количестве запускаемых в задаче одновременно параллельных процессов. Поскольку основное назначение ПО Slurm – запуск параллельных программ, использующих технологию MPI [4] для взаимодействия процессов параллельных программ, то в ПО Slurm используется для этого понятие «task» из MPI, хотя ПО Slurm также способно учитывать потоки OpenMP («threads») [5].

В эту же группу входят дополнительные параметры, уточняющие требуемые характеристики узлов: число сокетов для процессоров на узле, число ядер процессора, число потоков, предполагаемых к запуску на процессе, наличие и тип ускорителя, требуемую для процессов задачи оперативную память на узле, специальные возможности (features), назначенные администратором для вычислительных узлов и многое другое. Многие из этих параметров взаимосвязаны, это позволяет пользователю задать требуемую информацию максимально удобным способом.

Планировщик контроллера при просмотре очереди задач обращается к менеджеру ресурсов для подбора множества узлов, которые были бы свободны в настоящее время и удовлетворяли заданным в задаче требованиям. После успешного планирования задачи, т.е. в момент, когда необходимый набор узлов сформирован, менеджер ресурсов закрепляет их за задачами. После завершения процессов задачи менеджер ресурсов освобождает ресурсы, выделенные процессам, делая их доступными для других задач, ожидающих своей очереди.

4. Возможности базовой версии SLURM по управлению планированием запуска и запуском задач

Будучи, как следует из названия, в основном менеджером ресурсов, ПО Slurm, тем не менее, представляет возможности для планирования запуска задач. Таких возможностей три:

- использование одного из двух простых внутренних планировщиков;
- использование планировщика сторонних производителей, интегрируемых с ПО Slurm (например, Maui [6], Moab [7]);
- написание собственного планировщика (для этого предусмотрен уровень интерфейсов).

В состав ПО Slurm входят два типа внутренних планировщиков:

- builtin (встроенный);
- backfill (с обратным заполнением).

В общем виде в процессе планирования в ПО Slurm выполняются следующие действия:

- создается список готовых к запуску задач (т.е. таких, для которых выполнены все условия, например, время начала выполнения, и зависимости от состояния других задач);
- список задач упорядочивается по нескольким критериям, основным из которых является приоритет;
- если количество задач в списке превышает заданный предел, отсекается менее приоритетная часть списка;
- для каждой из задач формируется список ресурсов (разделов или резерваций), на которых допустим запуск этой задачи;
- предпринимается попытка вычисления времени запуска задач из списка.

Встроенный планировщик просто берет самую приоритетную задачу из нижней части списка и пытается подобрать для нее узлы из списка ресурсов, составленных для задачи. Если необходимые узлы находятся, задаче устанавливается текущее время для запуска и рассматривается возможность запуска следующей задачи. В противном случае процесс рассмотрения данного списка задач прекращается.

Планировщик с обратным заполнением пытается составить расписание запуска задач, учитывая возможность одновременного запуска нескольких задач. За основу опять же берется упорядоченный по приоритетам список задач, для которых с учетом заказанного времени счета строится расписание запуска. Затем свободные ресурсы заполоняются менее приоритетными задачами. При этом обеспечивается более плотное, по сравнению с встроенным планировщиком, заполнение вычислительного поля ВВС. При составлении расписания учитываются размер задачи (количество требуемых для ее выполнения узлов), заказанное время счета, возможность совмещения разных задач на узлах и многое другое. Приоритеты назначаются задачам динамически, исходя из времени нахождения задачи в состоянии ожидания, размеров задачи, заказанного времени счета.

Готовая к выполнению задача отправляется контроллером к компоненте slurmd на первом из выбранных для задачи узлов. Демон slurmd обеспечивает запуск процесса, на котором исполняется сценарий задачи, собирает и передает контролеру выводные файлы процессов задачи (stdout и stderr), контролирует использование внутренних ресурсов узла (CPU и оперативной памяти), а также отслеживает завершение задачи и передает это событие контроллеру.

5. Возможности базовой версии SLURM по расширению функций

Программное обеспечение Slurm ориентировано на использование подключаемых модулей (плагинов). Существует несколько типов плагинов и несколько реализаций каждого типа. Наиболее важные из них следующие:

- TaskPlugin – используется компонентой slurmd на узлах ВВС для запуска задачи и контроля за его исполнением;
- JobSubmitPlugin – используется утилитой sbatch (salloc) для проверки правильности заданных параметров и формировании на их основе атрибутов задачи;
- SchedPlugin – используется для расчета времени запуска задач;
- SelectPlugin – используется для подбора подходящих для задачи узлов и, при необходимости, запуска задачи на счет;
- PriorityPlugin – используется для вычисления приоритетов задач;
- JobAcctGatherPlugin – используется для подсчета использованных задачей ресурсов;
- группа плагинов под общим названием AcctGather – используется для подсчета используемых ресурсов ВВС (сетевой трафик, обращение к системе хранения и т.д.);
- PreemptPlugin – используется для определения возможности предоставления ресурсов задаче за счет вытеснения других задач (под приостановкой понимается временное прекращение счета задачи без освобождения занимаемых ей ресурсов оперативной памяти узлов);
- LaunchPlugin – используется для выполнения определенных действий при запуске задачи;
- JobCompPlugin – используется для выполнения определенных действий при завершении задачи.

Использование конкретной реализации плагина позволяет обеспечить более эффективное выполнение требуемых действий.

6. Анализ требуемых для ВВС функциональных особенностей и возможности их реализации в базовой версии SLURM

Базовые возможности ПО Slurm позволяют применять это ПО на самых различных ВВС. В тоже время при проведении массово-параллельных расчетов в научных организациях, в которых количество насчитываемых задач в год может превышать миллионные значения, а количество имеющихся вычислительных ресурсов не соответствует потребностям в вычислениях, остро встает вопрос о повышении эффективности использования вычислительных ресурсов имеющихся в наличии ВВС.

В таких случаях появляется понятие приоритетности задачи, т.е. появляется класс задач, который необходимо решить в первую очередь, даже, вполне возможно, за счет вытеснения с вычислительного поля ВВС уже запущенных в счет задач. Причем приоритетность задачи – величина, во-первых, непостоянная, во-вторых, она может являться результатом весовой функции из некоторого числа переменных. Критериями оценки значимости этих переменных в весовой функции могут быть важность решаемой конкретной задачи или группы задач, важность набора задач определенного пользователя или группы пользователей, объединенных общей научной или производственной направленностью.

Учитывая особенности проведения работ, в принципе в любой научной организации, подобной РФЯЦ-ВНИИТФ, для правильной организации расчетов можно выделить тематические вычислительные направления, связанные либо с исследуемыми физическими объектами, явлениями и процессами, либо с используемыми при расчетах математическими методами. Набор задач, решаемых по каждому из таких направлений, можно очень условно называть «методикой». Авторы публикации не претендуют на правильность и объективность данного термина, но такая терминология исторически была принята в РФЯЦ-ВНИИТФ.

Важность такого деления на наборы задач напрямую связана с понятием их приоритетности. Любая «методика» в какой-то момент времени, когда требуется срочно получить результат расчета, может стать приоритетной и наоборот, по факту получения результатов важность ее снижается. Таким образом, понятие «методика» может определять группу задач, обладающих общим признаком приоритетности и оперируя сочетанием двух понятий: «методика» – «ее приоритетность» в очереди задач планировщика можно более эффективно использовать ресурсы ВВС. В то же время внутри одной методики также, иногда, возникает необходимость выделения отдельных задач по уникальным для них показателям приоритетности.

Такое логическое объединение вычислительных задач на методики потенциально дает возможность задавать квоты по использованию ресурсов ВВС для конкретной группы задач. Что тоже, безусловно, позволяет более эффективно управлять ресурсами ВВС.

Помимо понятия приоритетности при расчетах на ВВС также целесообразно вводить определение «дневного» и «ночного» счета, связанные с активностью пользователей в рабочее время и, как следствие, определенной зависимостью прироста количества задач от времени суток. Смысл такого деления состоит в том, что в «ночном» режиме, например, не требуется немедленного получения результата от запущенной в счет задачи, так как в ночное время этот результат некому обработать. В силу отсутствия пользователей не происходит запуска новых задач, что позволяет планировщику составлять расписание, оптимально использующее вычислительные ресурсы. Режим «ночного» счета целесообразно также применять в выходные и праздничные дни.

В ходе проведения расчетов на ВВС нередко возникают ситуации, когда в вычислительной очереди скапливается множество разнородных задач, как имеющих важное значение для получения каких-то результатов расчета, так и предназначенных для отладки новых программ или алгоритмов. Ценность последних может оказаться небольшой с практической точки зрения, но, тем не менее, необходимость проведения тестовых и отладочных расчетов тоже безусловно существует. Часть из таких задач требуют небольших вычислительных ресурсов и непродолжительное время счета, но, тем не менее, не имеют возможности запуститься на счет в течение длительного времени из-за занятости ресурсов другими расчетными задачами. Более оперативный запуск таких задач за счет временного использования ресурсов программ, находящихся в расчете в течение нескольких суток или даже месяцев и занимающих большое количество вычислительных узлов, не сказался бы существенно на относительном времени их расчета, но способствовал более эффективному и разностороннему использованию вычислительных ресурсов ВВС.

Еще одной ключевой особенностью эффективного использования вычислительных ресурсов ВВС является способность системы управления (планирования запуска задач) менять размер задачи (количество запускаемых процессов и используемых вычислительных узлов) в зависимости от степени загруженности задачами вычислительного поля. В то время, когда необходимо обеспечить минимальное время задержки запуска, следует уменьшить количество вычислительных ресурсов, занятых задачами переменного размера, обеспечивая пространство для вновь запускаемых задач. А в периоды, когда количество активных пользователей вычислительной системы невелико, лучше предоставить максимальный вычислительный ресурс, для их эффективной утилизации. Естественно, при этом задачи сами должны иметь возможность свободной масштабируемости, т.е. относится к так называемому минимаксному типу задач (для них строго определены минимальные и максимальные потребности в ресурсах), и не относится к критичным по времени счета. Базовые возможности ПО Slurm в настоящее время позволяют задавать переменный размер задачи, но не используют его при планировании.

Важно отметить, что практически всегда на практике существует определенный класс задач, который не попадает по своим признакам и возможностям ни в одну из вышеперечисленных категорий. Это так называемые исключительные задачи. Например, задачи с постоянной

приоритетностью, задачи не имеющие функции автоматического сохранения при перезапуске, задачи, выполняемые в среде эмуляторов и т.п. Они малочисленны, но тем не менее имеют право на существование и исключительные права при распределении ресурсов на ВВС. Как правило, такие задачи являются невытесняемыми.

Есть еще один класс задач, на которые следует обратить особое внимание – это задачи, запускаемые администраторами ВВС в процессе обслуживания (диагностики) функциональных подсистем ВВС. Потребность в наиболее срочном прохождении этих задач очевидна. Возможность срочного запуска задач в этом случае достигается двумя способами. В случае, если требуется тестирование и наладка изолированного множества вычислительных узлов, то после их вывода из состояния наладки менеджер ресурсов автоматически распределяет их в резервацию, предназначенную для запуска тестов. В противном случае весь счет на ВВС, как правило, должен быть остановлен, и планировщик должен уметь работать с так называемым «белым списком» задач, которым предоставлено исключительное право запуска на вычислительном поле, тогда как другие пользователи могут продолжать постановку расчетных задач в очередь.

Коллективы пользователей ВВС, обслуживающие расчеты по различным методикам, могут быть достаточно велики, и организационно бывает удобным выделить из их состава администраторов расчета. В круг их возможностей должны входить функции снятия и перезапуска задач, а также управление порядком запуска задач собственной методики.

Функции, определяющие приоритет задачи, а также порядок рассмотрения кандидатов для вытеснения, зависят от многих критериев (занятость вычислительных ресурсов, среднее время счета задачи, характерное время сохранения данных для перезапуска и т.п.) и требуют аккуратной настройки весовых коэффициентов под каждый вычислительный центр и круг решаемых им задач.

Приведенные требования показывают, что базовых возможностей ПО Slurm явно недостаточно, чтобы удовлетворять столь разносторонним потребностям в управлении расчетами. Имеющиеся на рынке и в свободном доступе решения типа Maui или Moab реализуют часть вышеуказанных потребностей, но в основном в сочетании с менеджером ресурсов Torque [8]. Один из них – проект Maui давно прекратил свое развитие, другой, созданный на основе первого – Moab HPC Suite объединил в себе множество инструментов для управления ВВС и является платным продуктом, исходные коды которого закрыты и не позволяют их адаптировать под любые другие условия использования. Для ПО Slurm оба этих планировщика являются внешними, а протокол взаимодействия не обеспечивает достаточное количество событий и параметров. Поэтому в РФЯЦ-ВНИИТФ было принято решение расширить возможности ПО Slurm собственными разработками.

7. Расширенные возможности ПО SLURM-ВНИИТФ

7.1 Расширенная политика планирования задач

На основе проведенного анализа можно сделать выводы о том, какой может быть политика планирования задач для типовой ВВС, предназначенной для решения задач численного моделирования. Прежде всего, следует отметить, что эта политика подразумевает автоматическое управление запуском задач, и не требует участие администратора ВВС в процессе планирования.

Данная политика была реализована специалистами РФЯЦ-ВНИИТФ в ПО Slurm-ВНИИТФ. В ней задачи по значимости разделяются на следующие классы:

- интерактивные – короткоживущие задачи с минимальным требованием к вычислительным ресурсам, необходимых для отладки и визуализации результатов расчета;
- срочные – требующие как можно более быстрого решения, возможно, за счет прочих

задач;

- обычные – средний класс задач, не предъявляющих особых требований по срокам расчета;
- низкоприоритетные (фоновые) – класс ресурсоемких задач, решение которых представляет интерес, но срочность получения результатов от них не велика, поэтому такие задачи могут быть просчитаны после остальных или при появлении свободны вычислительных ресурсов, а будучи уже запущены, могут быть приостановлены (вытеснены из очереди задач) для предоставления ресурсов более приоритетным задачам.

Задача может быть причислена к соответствующему классу как индивидуально, так и по принадлежности к определенной группе задач. Предполагаются следующие типы группировки:

- по темам – в группу попадают все задачи, объединенные общей темой расчетов, например, расчет какой-либо конструкции, физического объекта, его модуля и т.д.
- по методикам – в группу попадают все задачи, проводящие расчеты по одной математической методике или их некоторому набору.

Способы группировки задач могут быть и другими, но в всех случаях для группировки используются данные из атрибутов задачи.

Для решения проблемы обеспечения высокоприоритетных задач вычислительными ресурсами были определены следующие механизмы:

- предоставление зарезервированных ресурсов,
- квотирование групп задач – для этого каждой группе назначаются ограничения на использование вычислительных ресурсов (лимиты), при превышении которых блокируется постановка новых задач той же группы в очередь на исполнение, либо такие задачи причисляются к классу фоновых,
- вытеснение менее приоритетных – в этом случае при появлении в очереди на выполнение срочной задачи, ей предоставляются ресурсы, занимаемые фоновыми задачами; вытесняемые задачи исключаются из счета и могут быть продолжены при появлении достаточного количества вычислительных ресурсов.

Для реализации этой политики были расширены функциональные возможности базовой версии Slurm:

- реализованы дополнительные возможности для задания атрибутов задач, позволяющих определить степень срочности задачи,
- реализованы различные способы группировки задач по их атрибутам,
- усовершенствованы алгоритмы планирования с учетом принадлежности задач к определенным группам, установленным квотам, степени приоритетности задачи, а также возможности вытеснения менее приоритетных задач,
- введена весовая функция расчета приоритетов,
- введена весовая функция пригодности задач к вытеснению,
- реализовано вытеснение фоновых задач для счета срочных задач,
- обеспечена возможность гибкого управления параметрами, определяющими политику планирования.

7.2 Описание структуры расширенной версии ПО SLURM-ВНИИТФ

Для реализации расширенных возможностей специалистами РФЯЦ-ВНИИТФ были разработаны следующие плагины:

- submit – постановка задачи в очередь, изменение параметров;
- priority – назначение приоритетов задачам;
- sched – расчет приоритетов задач;
- preempt – выбор задач для вытеснения;

- proctrack – передача сигналов процессам задач.

Работа плагинов управляется общим конфигурационным файлом формата YAML [9], разбитым на секции. Каждая секция содержит данные, необходимые для работы одного или нескольких плагинов. Отдельная нить контроллера отслеживает модификацию конфигурационного файла и перечитывает его содержимое, обеспечивая, тем самым, оперативное вмешательство администраторов в работу ПО Slurm-ВНИИТФ.

На рис. 2 приведен пример конфигурационного файла плагинов для ВВС.

```
# режим планирования
sched:
  mode: normal  # day или night взависимости от времени суток и дня недели
  blacklist: false  # режим черного списка (запрет запуска избранных)
  use_preemption: true
  use_limits: true
  interval: 30            # интервал планирования
  worktime: "07:00-19:00" # время работы в режиме встроенного планировщик
  workdays: [2,3,4,5,6]   # рабочие дни недели (первый день недели ВС)

  preempt: # секция вытесняющего планирования
    background_prio:     10
    standard_prio:    10000
    urgent_prio:    1000000

    urgent: # срочные задачи, не подчиняющиеся ограничениям
      - { name: '*.urgent' }
    background: # принудительно фоновые задачи
      - {name: '*.*.Ф*'}
    uncancelable: # невытесняемые задачи
      - { name: '*.*.Н*' }
    limits: # описание ресурсов и лимитов
      - {key: 'Quser1-1', user: User1,     use: PLMEM,   limit: 20}
      - {key: 'Quser1-2', user: User1,     use: PSMEM,   limit: 60}
      - {key: 'Quser2-1', user: User2,     use: PLMEM,   limit:  0}
      - {key: 'Quser2-2', user: User2,     use: PSMEM,   limit: 20}
# постановка в очередь
submit:
  default_part: "PSMEM"       # раздел для всех задач по умолчанию
  allow: # допустимые разделы (name, user, users, size -> partitions)
    - group: "Grp1"       # и группа пользователя Grp1
      partitions:         # список допустимых разделов
      - "*"               # допустим любой раздел
    - group: "Grp2"       # и группа пользователя Grp2
      partitions:         # список допустимых разделов
      - "PSMEM"           # допустим только раздел PSMEM
# приоритеты
prio:
  # по типу
  - {name: "*.П*", prio: 20}
  - {name: "*.М*", prio: 10}
  - {name: "*.Т*", prio:  8}
# администраторы методик
adm:
  - name: 'МЕТОДИКА1.*'
    admins:
      - {user: User1}
```

Рис. 2. Пример конфигурационного файла плагинов
Fig. 2. Configuration plugin example

Эта конфигурация содержит 4 основных секции (sched, submit, prio и adm), определяющие порядок планирования, постановку задачи в очередь, установку приоритетов и администраторов методик, соответственно. В данном примере задано чередование дневного и ночного режима планирования, определены признаки обнаружения срочных задач (окончание «.urgent» в имени задачи), фоновых задач (буква «Ф» в имени задачи), невытесняемых задач (буква «Н» в имени задачи). Кроме того, для пользователей установлены ограничения (квоты) на количество одновременно выделяемых узлов в разделах, установлено ограничение на использование разделов для разных групп пользователей и заданы приоритеты для разных типов задач. Для задач методики

«МЕТОДИКА1» назначен администратор User1, способный управлять задачами данной методики.

7.2.1 Типы задач

Для целей оптимального планирования пользователь должен указать тип задачи. Основными являются:

- производственная – задача по проведению расчетов в рамках плана работ предприятия;
- методическая – задача, связанная с уточнением существующих и отработкой перспективных вычислительных методик;
- тестовая – задача, связанная с тестированием оборудования или вычислительных методик.

В расширенной версии Slurm-ВНИИТФ также введены дополнительные типы задач.

- *Стандартные*: для каждой методики вводится ограничение на общее число вычислительных узлов, занимаемых суммарно всеми стандартными задачами данной методики, находящимися в счете. Сверх этого лимита стандартные задачи становятся фоновыми. Стандартной может быть любая задача с типами производственная, методическая, тестовая.
- *Фоновые*: в эту категорию попадают стандартные задачи, выходящие за пределы квоты. Они могут сниматься с расчета для освобождения вычислительных ресурсов. Объем вычислительных ресурсов, планируемых для расчета фоновых задач каждой методикой, ограничивается лишь количеством свободных вычислительных узлов, удовлетворяющих ограничениям задачи. Для реализации данного режима расчета предполагается, что задачи обязательно должны иметь возможность записи контрольных точек и продолжения счета с контрольной точки. Тип «фоновая» присваивается при постановке задачи в очередь на основании указанных атрибутов, может быть явно указан пользователем. При появлении нужной квоты возможен обратный переход из фоновой в стандартную.
- *Принудительно фоновые*: задачи, которые никогда не смогут стать стандартными. Их приоритет всегда будет находиться в диапазоне фоновых задач.
- *Отладочные и интерактивные*: данный тип предназначен для оперативного запуска задач, требующих небольших вычислительных ресурсов, с целью отладки программного кода в реальном окружении (на вычислительном поле). Данный тип присваивается тем задачам, время выполнения которых ограничено несколькими минутами и небольшим количеством вычислительных узлов.
- *Срочные*: задачи с приоритетом ниже отладочных, но выше чем диапазон стандартных задач. Для их запуска должны: автоматически сниматься фоновые задачи, автоматически сниматься со счета стандартные задачи этой же методики, автоматически прекращаться запуск стандартных задач. С расчета срочные задачи не снимаются никогда. Список срочных задач задается администратором в конфигурационном файле.
- *Невытесняемые*: задачи, которые не должны перезапускаться для запуска срочных и стандартных задач. Указывается пользователем при постановке задачи в очередь.

Плагин Priority на основании информации из секции prio конфигурационного файла, а также уточняющей информации из секций sched.preempt и sched.debug, рассчитывает начальное значение приоритетов задач. В вычислении приоритета участвуют тип задачи, ее размер и прочие атрибуты, заданные администратором.


```
sched:
  preempt:
    background_prio:     10
    standard_prio:    10000
    urgent_prio:    1000000
    urgent:                 # срочные задачи
      - { name: '*.urgent' }
    background:             # принудительно фоновые задачи
      - {name: '*.*.ф*'}
    uncancelable:           # невытесняемые задачи
      - { name: '*.*.Н*' }
  debug:
    use: true
    limit:   1
    time:   10
    size:    4
    priority: 1001000
prio:
  - {name: "*.П*", prio: 20}
  - {name: "*.М*", prio: 10}
  - {name: "*.Т*", prio:  8}
```

Рис. 3. Пример конфигурационного файла Priority
Fig. 3. Priority plugin example

Например, конфигурация, представленная на рис. 3, определяет:

- признаки производственных, методических и тестовых задач (буквы «П», «М» и «Т» в имени задачи) и их поправки к приоритетам (в целочисленном выражении: +20, +10 и +8, соответственно);
- признак фоновых задач (буква «Ф» в имени задачи) и базовый приоритет для них (10);
- признак невытесняемых задач (буква «Н» в имени задачи);
- базовые приоритеты для фоновых, стандартных и срочных задач;
- признак срочных задач (окончание «.urgent» в имени задачи) и приоритет их (1000000);
- признак отладочных задач (одна задача на пользователя с заказанным временем 10 минут и запрашивающая не более 4 узлов) и приоритет для них (1001000).

Полученные для задачи по разным критериям приоритеты складываются.

7.2.2 Режимы планирования

В расширенной версии Slurm-ВНИИТФ введены следующие режимы функционирования.

- *Дневной*: основная стратегия минимизирует время ожидания приоритетных задач в очереди. Обеспечивает быстрый запуск интерактивных и отладочных задач. В дневном режиме невозможен запуск задачи с низким приоритетом, пока в очереди находится высокоприоритетная задача.
- *Ночной*: минимизирует простои вычислительных ресурсов, характеризуется малой динамикой, возможностью составления долговременного расписания и более широкими возможностями оптимизации. Использует ресурсы, предоставленные днем для интерактивных и отладочных задач. В ночном режиме возможен запуск низкоприоритетных задач, в случае, если их счет не помешает запуску высокоприоритетных задач в запланированное время. Определение интервала времени для ночного режима задается в конфигурационных файлах Slurm.
- *Останов*: полный останов запуска задач. Используется во время профилактики для запрета запуска задач.
- *Режим белого списка*: запрет запуска задач кроме заданных категорий. Используется во время профилактики для тестирования вычислительной системы.
- *Режим черного списка*: запрет запуска задач заданных категорий. Этот режим совместим со всеми другими режимами планирования. Можно использовать для запуска срочных

задач за счет пространства заданной методики из черного списка.

Режимы планирования задаются в секции sched конфигурационного файла, пример такой конфигурации представлен на рис. 4:

```
sched:
  mode: day  #  установка дневного режима

sched:
  mode: night # установка ночного режима


sched:
  stop  #  установка признака прекращения планирования

sched:
  exclusive
exclusive:
           -{user:User1}  #  установка  разрешения  на  выполнение
           задач только пользователя User1

sched:
  blacklist
ban:
            -{user:User1} # установка  на запрет выполнения задач
            пользователя User1;
```

Рис. 4. Варианты выбора режима планирования в конфигурационном файле
Fig. 4. Scheduling mode variants in configuration file

Конфигурация, представленная на рис. 5, устанавливает чередование дневного и ночного режимов, днем считается промежуток времени от 7 до 19 часов в рабочие дни:

```
sched:
   mode: normal
   worktime:  "07:00-19:00"   #время  работы  в  режиме  встроенного
планировщика (формат XX:XX-XX:XX)
   workdays: [2,3,4,5,6]    #рабочие дни недели (первый день недели
ВС)
```

Рис. 5. Пример раздела sched конфигурационного файла Priority
Fig. 5. Example of the sched section in the configuration file Priority

7.2.3 Правила именования задач и задание информации о начислении ресурсов

Имя задачи является одним из дополнительных источников информации и должно соответствовать шаблону МЕТОДИКА.ЗАДАВА.ТИП[.ОПИСАНИЕ].

Поле «МЕТОДИКА» должно содержать имя «методики», известной Slurm.

Поле «ЗАДАВА» должно содержать числовой номер задачи и ее варианта.

Поле «ТИП» может быть одним из представленных ниже:

- П – производственная;
- М – методическая;
- Т – тестовая;
- Ф – фоновая;
- Н – невытесняемая.

Поле «ОПИСАНИЕ» может содержать произвольную дополнительную информацию о задаче (в том числе ту, которую можно дополнительно применять при расчете приоритетов или принятия решения об исключительности задачи).

Информация о начислении ресурсов, (т.е. как их регистрировать в статистике) задается в атрибуте задачи «ACCOUNT» по шаблону ТЕМА:ЗАКАЗЧИК, где «ТЕМА» - идентификатор темы, а «ЗАКАЗЧИК» – идентификатор заказчика. И то, и другое содержится в отдельных файлах.

Указанная информация обрабатывается плагином Submit. По информации, заданной в секции submit конфигурационного файла плагинов, этот плагин проверяет правильность заданной информации и, при необходимости, отвергает задачу или формирует недостающие атрибуты задачи или переопределяет заданные пользователем атрибуты на указанные администратором.

7.2.4 Группы задач

Множество параметров, используемых при планировании, применяются к группам задач, объединенных общими признаками. Для выделения требуемой группы из всего множества задач применяются фильтры. Каждый фильтр сравнивает один из атрибутов задачи с заданным значениям. Фильтры применяются последовательно, в порядке их написания в конфигурационном файле.

```
name: '*.*.П.*'
group: Grp1
```

Рис. 6. Пример фильтра группы задач
Fig.6. Example of a task group filter

Например, фильтр, представленный на рис. 6 выбирает все производственные задачи группы «Grp1», а фильтр, представленный на рис., выбирает все производственные задачи методики «МЕТОДИКА1» пользователя «User1».

```
name: 'МЕТОДИКА1.*.П.*'
user: User1
```

Рис. 7. Пример фильтра группы задач
Fig. 7. Example of a task group filter

7.2.5 Распределение ресурсов

Для каждой группы задач выделяется квота на использование ресурсов. Основным ресурсом ВВС являются узлы вычислительного поля, поэтому для групп задач указывается максимально допустимое число узлов в каждом разделе. Ресурсы вычислительного комплекса делятся между методиками таким образом, чтобы небольшая часть ресурсов оставалась свободной для отладочных и интерактивных задач. Стандартные задачи могут использовать только выделенную группе задач квоту. Сверх этой квоты могут существовать только срочные или фоновые задачи. Таким образом, у каждой группы задач есть свое выделенное количество серверов, на которые могут претендовать только задачи этой группы (за исключением срочных и фоновых задач).

```
sched:
  preempt:
    limits:
      - {key: 'M1-PL', name: 'Методика1', use: PLMEM, limit: 80}
      - {key: 'M1-PS', name: 'Методика1', use: PSMEM, limit: 20}
      - {key: 'M2-PS', name: 'Методика2', use: PSMEM, limit: 60}
```

Рис. 8. Пример задания квот в файле конфигурации
Fig. 8. Example of setting quotas in a configuration file

Квоты задаются в секции sched.preempt.limits конфигурационного файла. Например, то, что представлено на рис. 8, указывает, что задачи методики 1 могут использовать в разделе PLMEM одновременно все 80 узлов, а в разделе PSMEM не более 20; методика 2 к разделу PLMEM вообще не допущена, а в разделе PSMEM может использовать не более 60 узлов. В этом примере используются фильтры по имени задачи и заказанного раздела.

Другой пример, где вместо методик используется фильтр по группам, представлен на рис. 9.

```
sched:
  preempt:
    limits:
      - {key: 'G1-PL', group: 'Grp1', use: PLMEM, limit: 80}
      - {key: 'G1-PS', group: 'Grp1', use: PSMEM, limit: 20}
      - {key: 'G2-PS', group: 'Grp2', use: PSMEM, limit: 60}
```

Рис. 9. Пример задания квот по группам в файле конфигурации
Fig. 9. Group quota example

Установленные квоты снабжаются уникальными идентификаторами (key). Текущее значение установленных и использованных квот можно в любой момент посмотреть утилитой qinfo, как представлено на рис. 10.

```
$ qinfo
Режим              : ночной (backfill)
Квоты              : включено
Вытеснение         : включено
Свободно BУ        :   25
Свободно PLMEM     :   20
Свободно PSMEM     :    5
M1-PL              : ограничение 80 использовано  60 доступно 20
M2-PS              : ограничение 20 использовано  20 доступно  0
M2-PS              : ограничение 60 использовано  55 доступно  5
$
```

Рис. 10. Использование утилиты qinfo для просмотра информации о квотах
Fig. 10. Utility qinfo output example

7.2.6 Администрирование групп задач

Базовая версия Slurm не предоставляет пользователем никаких инструментов для управления задачами других пользователей. Пользователь может вмешиваться в работу Slurm только двумя способами:

- удалять или перезапускать свои задачи;
- менять атрибуты своих задач командой scontol update job, косвенно влияя на приоритет задачи, задавая атрибут «nice» или переопределяя тип задачи в ее имени.

Однако выделение групп задач в отдельную категорию обрабатываемых плагинами данных и появление множества атрибутов, влияющих на порядок запуска задач, позволяет выделить отдельную категорию пользователей, которым можно делегировать модификацию этих атрибутов и частичного управления порядком запуска в пределах своей группы задач.

Таким образом, введено понятие «администратор группы задач» или «администратор методики». Список администраторов задается в секции adm конфигурационного файла. Например, так, как представлено на рис. 11.

```
adm:
  - name: 'МЕТОДИКА1.*.*'
    admins:
      - {'User1'}
  - name: 'МЕТОДИКА2.*.*'
    admins:
      - {'User2'}
```

Рис. 11. Пример задания функций администратора в конфигурационном файле ПО Slurm-ВНИИТФ
Fig. 11. Administrator function example in the Slurm-VNIITF configuration file

Приведенная на рис. 11 запись показывает, что пользователь «User1» является администратором методики «МЕТОДИКА1», а пользователь «User2» – администратором методики «МЕТОДИКА2». Плагин Submit на основании этой информации разрешает выполнение команды scontrol update job указанным пользователям для всех задач их групп. Таким образом, администратор группы задач может менять тип и приоритет задач в пределах своей группы.

7.2.7 Вытеснение задач

При появлении в очереди задачи, для которой не хватает ресурсов, может быть предпринята попытка освобождения ресурсов для нее путем перезапуска находящихся в счете менее приоритетных задач, необъявленных невытесняемыми.

Существует две стратегии вытеснения задач, в зависимости от типа задачи.

При появлении в очереди срочной задачи кандидаты на вытеснение ищутся в следующем порядке:

- сначала рассматриваются фоновые задачи, входящие в ту же группу, что и срочная,
- затем рассматриваются прочие фоновые задачи,
- в последнюю очередь рассматриваются стандартные задачи из своей группы.

При появлении в очереди стандартной задачи, укладывающейся в квоты группы, кандидаты на вытеснение ищутся из числа фоновых задач других групп, превысивших лимиты.

В других случаях вытеснение не производится, и задача ждет освобождения достаточных для ее запуска ресурсов.

После нахождения подходящих задач производится попытка их перезапуска. Выбранные задачи должны сохранить свое состояние и завершится, перейдя в состояние ожидания запуска. Поскольку срочная задача имеет наивысший приоритет, освободившиеся в результате ресурсы будут предоставлены ей. Позднее перезапущенные задачи будут снова автоматически запущены.

Предположим, что в конфигурационном файле задана следующая информация, представленная на рис. 12.

```
sched:
  preempt:
    urgent:
      - {name: 'МЕТОДИКА1.12345.П*'
    background:
      - {name: '*.*.Ф*'
    uncancelable:
      - {name: '*.*.Н*'
```

Рис. 12. Пример задания фильтра вытеснения задач в конфигурационном файле ПО Slurm-ВНИИТФ
Fig. 12. Task preemption filter example in the Slurm-VNIITF software configuration file

Тогда после появления в очереди задачи с именем «МЕТОДИКА1.12345.П.Расчет…», которая будет распознана как срочная, из списка фоновых задач (содержащих букву «Ф» в третьем слоге имени), находящихся в состоянии выполнения, но не являющихся невытесняемыми (т.е. в третьем слоге имени нет буквы «Н»), будут выбраны подходящие по размеру задачи. Если такие задачи будут найдены, им посылается сигнал о прекращении работы, получив который, фоновые задачи должны сохранить свое состояние и завершиться. После завершения, фоновые задачи остаются в очереди в состоянии ожидания запуска, а освобожденные ими ресурсы будут предоставлены срочной задаче.

Важно отметить еще один аспект. Иногда, на первый взгляд, достаточно подобрать для вытеснения такое множество фоновых задач, чтобы занимаемые ими ресурсы были минимально достаточными для счета высокоприоритетной задачи. Однако, стоимость возобновления счета фоновых задач разная, и, как правило, заранее известна для конкретной методики. Например, если фоновой задаче до ее завершения осталось времени меньше, чем потребуется на её перезапуск (время сохранения промежуточных данных расчетов и время считывания данных контрольной точки при запуске задачи существенны), то решение о перезапуске может быть нерациональным. В этом случае оптимальным решением будет отложить запуск срочной задачи до завершения расчета фоновой или выбрать другую задачу для вытеснения. Кроме того, фоновые задачи в очереди могут сильно отличаться по размеру запрашиваемых ими ресурсов. В этом случае время ожидания для задач большего размера растет нелинейно, так как вероятность освобождения малого количества вычислительных

ресурсов на ВВС во много раз больше, чем больших. Это значит, что предпочтительней выбирать для вытеснения несколько небольших задач, чем одну задачу, занимающую необходимые вычислительные ресурсы. Множество учитываемых при планировании очереди задач факторов также должно содержать количество перезапусков задачи, близость времени последнего перезапуска, и поправку к приоритету, которая может быть установлена администратором методики, самим пользователем, запускающим задачу или администратором ВВС.

Таким образом, в результате всех исследований, авторами публикации был разработан некий аналитический подход, когда для каждой вытесняющей задачи строится множество вытесняемых задач J_{ptee}, занимающих ресурсы, подходящие вытесняющей, как показано на рис. 13. Из этого множества формируется набор, элементами которого являются подмножества различных сочетаний вытесняемых задач MJ_{ptee} таких, что используемых ими суммарных ресурсов достаточно для запуска вытесняющей. Затем, для каждого элемента набора вычисляется функция стоимости вытеснения этого подмножества задач. Для вытеснения выбирается тот элемент набора, для которого оценочная функция возвращает минимальное значение.

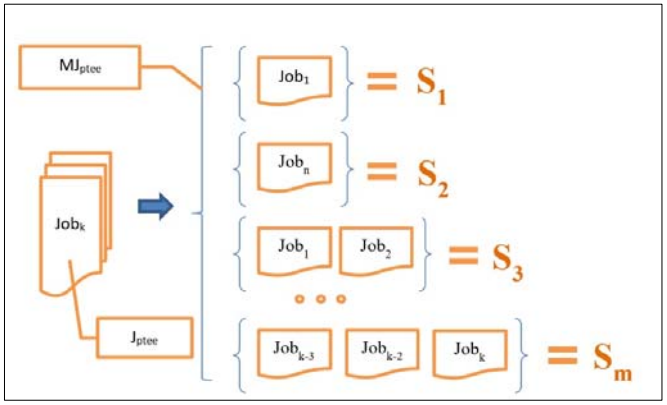

Рис. 13. Схема определения множества вытесняемых задач
Fig. 13. Preemptable task set definition scheme

В результате выбор множества вытесняемых задач для перезапуска сводится к минимизации функции стоимости вытеснения множества задач на интервал планирования Δt:

$$M = \min\{S_1, S_2, \dots, S_m\},$$

Здесь S_m– это вычисляемая на основе набора весовых функций и их коэффициентов функция стоимости вытеснения m-ого набора вытесняемых задач:

$$S_m = \sum_{i=1}^{n}(k_1 Tend_i + k_2 Tr_i + k_3 Tcalc_i + k_4 N_i + k_5 R_i + k_6 S_i + k_7 A_i),$$

где:

n – количество вытесняемых задач в подмножестве (т.е. рассматриваемом элементе набора);

$k_1 \cdots k_7$ – конфигурируемые коэффициенты факторов стоимости;

$Tend_i, Tr_i, Tcalc_i, N_i, R_i, S_i, A_i$ – функции факторов стоимости.

Рассмотрим подробнее каждую из весовых функций факторов стоимости.

$Tend_i$ – весовая функция времени завершения счета задачи:

$$Tend_i = c_1^{\left(\frac{t_i - now}{\Delta t}\right)},$$

где:

c_1 – переменный показатель степенной функции (0,5…0,6);

t_i – предполагаемое время завершения задачи;

now – текущее время;

Δt – интервал вытеснения.

Tr_i – весовая функция времени последнего перезапуска задачи:

$$Tr_i = c_2^{\frac{now - tr_i}{\Delta t}},$$

где:

c_2 – переменный показатель степенной функции (0,2…0,3);

tr_i – время последнего перезапуска задачи;

now – текущее время;

Δt – интервал вытеснения.

$Tcalc_i$ – весовая функция полного времени счета задачи:

$$Tcalc_i = c_3^{\frac{now - tcalc_i}{\Delta t}},$$

где:

c_3 – переменный показатель степенной функции (0,2…0,3);

$tcalc_i$ – предполагаемое полное время счета задачи;

now – текущее время;

Δt – интервал вытеснения.

N_i – весовая функция количества перезапусков задачи:

$$N_i = 1 - \frac{1}{2}^{rc_i},$$

где:

rc_i – количество перезапусков задачи в процессе счета.

R_i – весовая функция возможности перезапуска задачи:

$$R_i = \begin{cases} r_i = 1 \\ r_i = 0, \end{cases}$$

где:

r_i – возможность перезапуска, определяемая пользователем, администратором методики, администратором ВВС.

S_i – весовая функция размера задачи:

$$S_i = 1 - c_4^{s_i},$$

где:

c_4 – переменный показатель асимптотической функции (0,996…0,997);

s_i – размер задачи в вычислительных узлах.

A_i – весовая функция поправки приоритета.

$$A_i = 1 - c_5^{n_i},$$

где:

c_5– переменный показатель асимптотической функции (0,993…0,995);

n_i – поправка к приоритету, задаваемая администратором методики или пользователем.

Коэффициенты $k_1 \cdots k_7$ так же, как и показатели функций $c_1 \dots c_4$, представляют собой значения, которые определяются эмпирически, на основе большого набора статистических данных и зависят в свою очередь от размерности ВВС, количества просчитываемых на ней ежегодно задач, многообразия типов задач.

8. Заключение

Использование разработанной в РФЯЦ-ВНИИТФ версии ПО Slurm-ВНИИТФ позволило существенно расширить возможности проведения расчетов на ВВС, увеличить эффективность использования имеющихся вычислительных ресурсов. Благодаря ПО Slurm-ВНИИТФ удалось увеличить ежегодное количество просчитываемых задач примерно на 20%. В то же время набор реализованных функций и удобный, интуитивно понятный интерфейс по их настройке, существенно упростили работу администраторов ВВС, позволив им оперативно влиять на очередь задач в зависимости от текущих потребностей в расчетах различных подразделений института.

Результаты данной работы носят, безусловно, достаточно общий и субъективный характер и отражают один из возможных подходов к организации проведения расчетов задач численного моделирования на ВВС. Тем не менее, авторы считают, что опубликованная ими работа может быть полезна специалистам, занимающимся разработкой специализированного ПО для подсистем управления ВВС. Авторы намерены продолжить публикации по данной тематике, с целью представления более развернутой информации по каждой из функциональных подсистем ВВС, основным аспектам и особенностям администрирования ВВС, особенностям разработки прикладного и системного ПО для ВВС, предназначенных для решения задач численного моделирования.

Список литературы / References

[1] Игнатьев А.О., Мокшин С.Ю. Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования, Препринт РФЯЦ-ВНИИТФ № 265, Снежинск, 2020 г., 21 с. / Ignatyev A.O., Mokshin S.Yu. Base architecture of the mathematical modelling HPC system, Preprint FSUE «RFNC-VNIITF named after Academ. E.I. Zababakhin» № 265, Snezhinsk, 2020, 21 p. (in Russian).

[2] «СПО Супер-ЭВМ», Available at: http://vniitf.ru/article/spo-super-evm, accessed 01.04.2022 (in Russian).

[3] Slurm workload manager, Available at: https://slurm.schedmd.com/documentation.html, accessed 01.04.2022.

[4] MPI: The Message Passing Interface. Available at: http://parallel.ru/tech/tech_dev/mpi.html, accessed 01.06.2020.

[5] The OpenMP API specification for parallel programming. Available at: https://www.openmp.org/, accessed 01.04.2022.

[6] Maui Scheduler, Available at: https://github.com/TempleHPC/maui-scheduler, accessed 01.04.2022.

[7] Moab Cluster Suite, Available at: https://adaptivecomputing.com/moab-hpc-suite/, accessed 01.04.2022.

[8] Torque Resource Manager, Available at: https://adaptivecomputing.com/cherry-services/torque-resource-manager/, accessed 01.04.2022.

[9] YAML, Available at: http://yaml.org /, accessed 01.04.2022.

Информация об авторах / Information about authors

Алексей Олегович ИГНАТЬЕВ – начальник лаборатории. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – Head of Laboratory. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

Алексей Алексеевич КАЛИНИН – начальник группы. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, разработка компиляторов, методы оптимизации планирования вычислений в параллельных системах, методы формальной верификации алгоритмов.

Alexey Alexeevich KALININ – Head of Research Group. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, compilers and debuggers development, task management optimization for HPC system, formal verification of algorithms methods.

Сергей Юрьевич МОКШИН – начальник отдела. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – Head of Department. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means of information security.