УДК 004.051 doi 10.26089/NumMet.v20r319

ЦЕЛЕВАЯ ОПТИМИЗАЦИЯ СТРУКТУРЫ ПОТОКА ЗАДАЧ СУПЕРКОМПЬЮТЕРОВ

С. Н. Леоненков[1]

Настоящая статья является результатом исследования потока задач суперкомпьютеров “Ломоносов” и “Ломоносов-2”. Предложен подход к оценке эффективности функционирования суперкомпьютерной системы, основанный на ее базовых характеристиках. Введена новая функция потери качества планирования суперкомпьютера. Подход позволяет сравнивать разные суперкомпьютерные системы исходя из целей использования, которые стоят перед системными администраторами вычислительных центров. Описан опыт применения целевой оптимизации процессов планирования, основанной на предложенном подходе, для суперкомпьютеров “Ломоносов” и “Ломоносов-2”.

Ключевые слова: суперкомпьютер, эффективность планирования потока задач, алгоритмы планирования, SLURM.

1. Введение. Суперкомпьютерные комплексы различают по целям эксплуатации, которые ставят перед собой их администраторы. Два показательных примера таких целей — это цели систем из списка TOP500: “Ломоносов-2” МГУ имени М.В. Ломоносова (79 место в TOP500 по состоянию на ноябрь 2018 г.) и “Mira” Аргоннской национальной лаборатории США (21 место в TOP500 по состоянию на ноябрь 2018 г.). Эти системы обладают колоссальным возможностями для запуска задач с высоким уровнем параллелизма, большой базой пользователей и обширным пулом решаемых ими задач. Однако цели обеих суперкомпьютерных установок кардинально отличаются. Перед системой “Ломоносов-2” стоит цель дать достаточный доступ максимальному количеству исследовательских и учебных групп, тогда как основная цель кластера “Mira” — предоставлять процессорное время для экстремально больших задач.

В последние годы, говоря об эффективности использования крупных вычислительных систем, обычно имеют в виду утилизацию их процессорного времени. Из года в год средняя эффективность, рассчитываемая таким образом, растет. Однако на примере систем “Ломоносов-2” и “Mira” понятно, что утилизация процессорного времени не будет объективно оценивать эффективность выполнения целей этих систем. В настоящей статье предлагается подход к оценке эффективности использования крупных вычислительных систем, основанный на выделении фундаментальных характеристик потока задач системы и представлении их комбинации в виде целевой функции, характеризующей показатель качества достижения целей, поставленных перед планировщиком задач. Этот подход позволяет сравнивать разные суперкомпьютерные комплексы несмотря на большое разнообразие целей процессов планирования их ресурсов.

Разработчики систем планирования ресурсов суперкомпьютеров используют различные подходы к достижению поставленных целей. Основными подходами в последние годы были алгоритмы FCFS (First Come First Served) и Backfill (и множество его оптимизаций). В настоящее время на рынке ПО выделяются менеджер ресурсов с открытым исходным кодом SLURM (Simple Linux Utility for Resource Management) и коммерческий планировщик IBM Spectrum LSF (Load Sharing Facility). По данным разработчиков SLURM [25], в 2017 г. этот менеджер использовали 6 из 10 самых производительных систем по версии списка TOP500.

Кроме алгоритмов планирования, для оптимизации структуры потока задач также используют дополнительные ограничения, которые накладывают системные администраторы, на доступ в поток задач, ожидающих постановку на исполнение. Например, существуют различные системы приоритетов, лимиты на количество задач в очереди и на их исполнение от каждого пользователя/группы, лимиты на запрошенные и занятые процессоро-часы и др.

Целью нашей работы является исследование и разработка методов анализа эффективности планирования и использования ресурсов крупных вычислительных комплексов, а ее задачей является разработка методов, подходов, алгоритмов и программных средств, направленных на оптимизацию структуры потока задач в соответствии с целями использования и заданными параметрами суперкомпьютерных систем.

[1] Московский государственный университет им. М.В. Ломоносова, факультет вычислительной математики и кибернетики, Ленинские горы, 119992, Москва; аспирант, e-mail: leonenkov@cs.msu.ru

Структура статьи устроена следующим образом: вторая часть посвящена обзору подходов и технологий повышения эффективности планирования потока задач, третья — рассматривает многоцелевой подход к планированию потока задач крупных вычислительных комплексов, четвертая часть описывает апробацию описанного подхода.

2. Обзор подходов и технологий повышения эффективности планирования потока задач суперкомпьютерных комплексов. В данном разделе представлен обзор инструментов (планировщиков) и подходов к планированию структуры потока задач суперкомпьютера, рассмотрены опыт работы и подходы к решению задачи повышения эффективности планирования крупнейших суперкомпьютеров.

2.1. Системы управления ресурсами суперкомпьютеров. Для достижения максимальной эффективности планирования ресурсов своих суперкомпьютерных установок системные администраторы используют большой спектр различного ПО (от внешних планировщиков до полновесных менеджеров ресурсов): Portable Batch System (PBS), TORQUE, PBS Professional, Loadleveler, LSF, SLURM, Maui, MOAB и др., а также отечественные СУППЗ (МСЦ) и Cleo (НИВЦ МГУ) [2]. В данный момент PBS доступен для администраторов в трех комплектациях: PBS Professional, TORQUE и OpenPBS. Самая популярная из них TORQUE [3] поддерживается компанией Adaptive Computing [6] и является логическим продолжением версии OpenPBS, которая уже давно не поддерживается. PBS Professional [26] — еще одна версия PBS, которую поддерживает компания Altair Engineering, кроме opensource комплектации, дополнительно поставляется коммерческая версия платформы [27]. Loadleveler [28] — планировщик заданий для крупных вычислительных комплексов, который поставляла компания IBM. В МГУ данный планировщик успешно эксплуатируется на системе IBM Blue Gene/P [7]. Последняя версия поддерживалась до 2016 г. [8], после чего компания предложила пользователям мигрировать на свою систему для управления кластерами LSF [9].

IBM LSF — коммерческий пакет компании IBM, в качестве основных особенностей можно выделить поддержку контейнеров (например, Docker), графический интерфейс взаимодействия с кластером для пользователей и расширенный мониторинг.

В НИВЦ МГУ разрабатывалась система управления заданиями Cleo [10]. Планировщик поддерживает систему очередей и приоритетов, некоторый предопределенный набор лимитов, блокировки и стратегии выбора процессоров. На данный момент нет известных суперкомпьютерных установок, которые продолжают использовать планировщик Cleo.

Еще одним примером удачного решения проблем планирования ресурсов больших вычислительных комплексов является система управления прохождением параллельных заданий (СУППЗ) [11]. Одной из главных опций, которая отличает СУППЗ от своих конкурентов, является механизм фоновых заданий [12]. Также доступна возможность подключения внешних планировщиков, например Maui.

Maui — планировщик заданий, который до 2005 г. активно использовался вычислительным сообществом [13, 14]. Его отличительными особенностями были разветвленная система приоритетов и качественный стек алгоритмов планирования ресурсов. После 2005 г. планировщик был трансформирован в коммерческий и стал называться MOAB [15]. Многие менеджеры ресурсов предоставляют API для использования Maui/MOAB в качестве внешних планировщиков [16]. Подробное сравнение ранних систем рассмотрено в статьях [2] и [17].

Основной системой планирования для крупнейших вычислительных комплексов мира на данный момент является SLURM [18]. SLURM успешно используется не только на большинстве самых производительных систем рейтинга TOP500, но и на суперкомпьютерах “Ломоносов” и “Ломоносов-2” (самая производительная система СНГ на момент публикации). Система до недавнего времени поддерживала API для взаимодействия с внешними планировщиками (плагины wiki и wiki2) [19]. SLURM активно расширяется и поддерживается, ежегодно выходят обновления. Доступна открытая версия и коммерческая поддержка от SchedMD LLC [20]. Менеджер SLURM создан для крупных вычислительных кластеров и поддерживает до 10 000 000 процессоров, отдельно надо отметить, что SLURM является достаточно быстродействующим и может запускать до 500 заданий в секунду. Основным конкурентным преимуществом SLURM является его модульность. Более подробное описание системы и пример его использования на суперкомпьютере “Ломоносов” рассмотрены в статьях [21] и [22].

2.2. Цели и подходы к планированию потока задач суперкомпьютерных комплексов. В области подходов к планированию ресурсов суперкомпьютеров доминируют алгоритмы типа Backfill. Сейчас существует множество оптимизаций этого алгоритма, в том числе основанные на предсказании методами машинного обучения времени работы отдельных задач пользователей. Кроме версий Backfill, до сих пор для удовлетворения разных потребностей системных администраторов используются стандартные алгоритмы: FCFS (First Come First Served), SJF (Shortest Job First), LJF (Longest Job First) и др.

Дополнительно системные администраторы применяют механизмы квотирования задач пользователей и проектов (например, по количеству задач или процессорному времени), системы приоритетов и разнообразные алгоритмы выборов процессоров. Многие вычислительные центры самостоятельно создают алгоритмы под свои цели планирования вычислительных ресурсов.

Кроме всего прочего, разные суперкомпьютерные центры ставят перед собой разные цели эксплуатации своих систем. Рассмотрим примеры целей следующих двух систем: “Ломоносов-2” из МГУ и “Mira” из Аргоннской национальной лаборатории.

Перед системой “Ломоносов-2” стоит цель: дать исследовательским и академическим группам пользователей доступ к высокопроизводительным параллельным вычислениям, причем предоставить этот доступ максимально возможному их количеству. Суперкомпьютерный центр МГУ на сегодняшний день предоставляет доступ более 3000 пользователям и 900 проектам одновременно. Более 1000 задач выполняются на установках центра (в том числе на суперкомпьютере “Ломоносов-2”) ежедневно. Благодаря тому факту, что “Ломоносов” [1] и “Ломоносов-2” — гетерогенные суперкомпьютеры, на обоих создана система очередей, которые позволяют пользователю наиболее эффективно выбрать вид вычислительных узлов, исходя из архитектуры своей программы.

В свою очередь, отделение Аргоннской национальной лаборатории — Argonne Leadership Computing Facility (ALCF) [23] — ставит перед собой следующую цель эксплуатации своих систем: утилизация вычислительного времени системы является важной целью в ALCF, но самая большая цель — это позволить заданиям экстремального масштаба максимально быстро запускаться [24]. С 2013 по 2017 г. суперкомпьютер “Mira” дал доступ к своим ресурсам более 550 проектам и 1000 пользователям, более 280 000 задач было запущено на этой системе. Для решения своей амбициозной цели ALCF отошли от традиционного подхода к планированию ресурсов кластера, они придумали свою “функцию утилизации” и реализовали специальную систему очередей, которая позволяет максимально эффективно решать поставленную задачу [24].

Отдельно следует упомянуть о нестандартном подходе к решению задачи повышения эффективности планирования потока задач для суперкомпьютера Titan. Разработчики программного обеспечения решили задачу простаивания ресурсов следующим образом: системные администраторы накопили большое количество однопроцессорных задач, которые могут быть быстро развернуты и свернуты, и запускают их на любых свободных ресурсах. Как только в очереди появляются задачи пользователей, которые претендуют на ресурсы, занятые такими задачами, то выполнение приостанавливается, однопроцессорные задачи сворачиваются и новый пользователь получает доступ к запрошенным процессорам. Таким образом, эффективность использования ресурсов Titan была максимально поднята.

Исходя из многообразия целей планирования задач крупных вычислительных систем, становится очевидно, что только характеристика утилизации системы (как самый популярный подход к определению эффективности суперкомпьютеров) не покрывает всех целевых функций эффективности планирования разных суперкомпьютеров. В связи с этим необходим комплексный подход, который дает возможность комбинировать различные цели и позволяет сравнивать между собой разные установки в соответствии с их задачами.

3. Многоцелевой подход к планированию потока задач крупных вычислительных комплексов. Чтобы подробнее описать задачу определения эффективности планирования потока задач суперкомпьютера, необходимо ввести дополнительные понятия: задание, функция потери качества упаковки, характеристика функционирования суперкомпьютера и др.

3.1. Задача определения эффективности планирования потока задач суперкомпьютера. Пусть задана полоса фиксированной ширины H, отражающая занятость ресурсов вычислительного комплекса во времени (H — число узлов кластера). На полосе задана система координат XY (координата X соответствует времени, Y — числу узлов). В полосе задается окно W длины T, определяющее отрезок времени. Под координатой начала окна понимается координата его левого нижнего угла (X_0, Y_0) (рис. 1).

Задание — это программа пользователя, которая может находиться в двух состояниях: либо стоять в очереди, либо исполняться на вычислительных ресурсах.

Определение 1. Представим задание как совокупность элементов $J_i = \{X_i, T_i, H_i, R_i, U_i, Q_i\}$, где:

— X_i — момент времени запуска задачи на исполнение на вычислительных ресурсах;

— T_i — время исполнения задания на вычислительных ресурсах;

— H_i — число запрошенных вычислительных узлов для исполнения задания;

— R_i — непустой набор из j пар вида (y_{ij}, h_{ij}), который описывает распределение задания по узлам

в виде прямоугольников с координатой левого нижнего угла в (Xi, y_{ij}), временем исполнения T_i и числом узлов h_{ij}, таким, что $\sum\limits_j h_{ij} = H_i$ (рис. 2 и 3);

— U_i — идентификатор пользователя, ассоциированного с заданием;

— Q_i — время постановки задания в очередь.

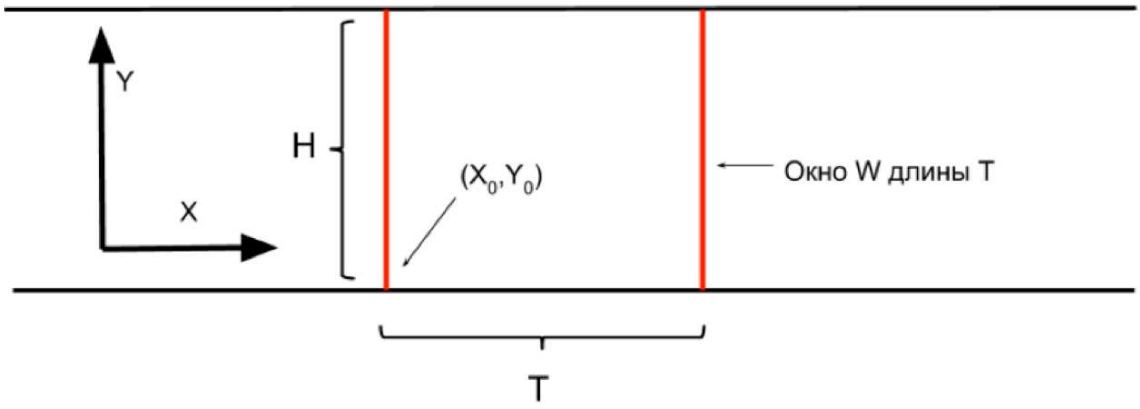

Рис. 1. Геометрическая интерпретация исполнения потока задач на суперкомпьютере. Полоса/Окно

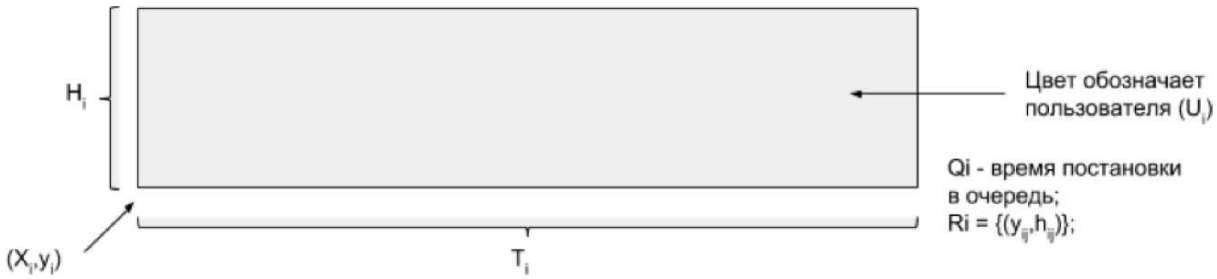

Рис. 2. Геометрическая интерпретация понятия задание

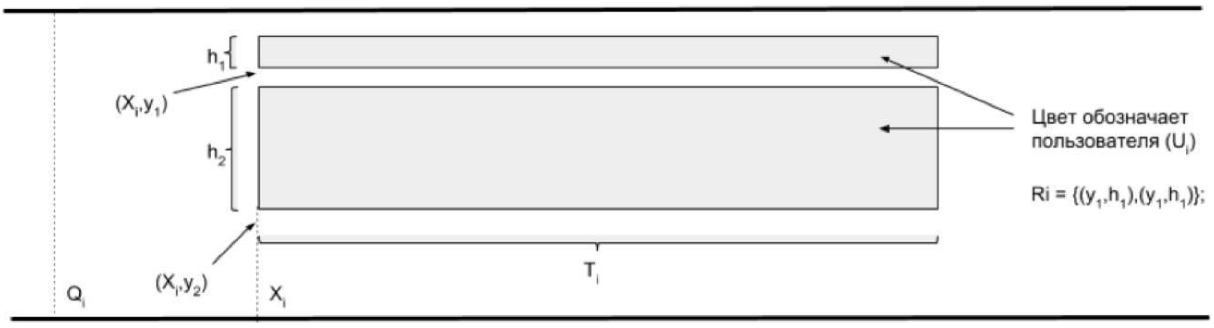

Рис. 3. Задание J_i в полосе и вариант разбиения R_i

Интерпретация обозначений. Задание соответствует прямоугольнику с заданными координатами левого нижнего угла, известными размерами (H_i и T_i — размеры прямоугольника по осям Y и X соответственно), заданным цветом (соответствует идентификатору пользователя) и разбиением по узлам R_i (рис. 2).

Рассмотрим два набора заданий:

— Z_{start} — набор выполняющихся в момент времени $X_i = X_0$ заданий (рис. 4);

— Z_{queue} — набор находящихся в очереди заданий, для которых не заданы координаты X_i и разбиение R_i, а $Q_i \leqslant X_0 + T$ (рис. 5).

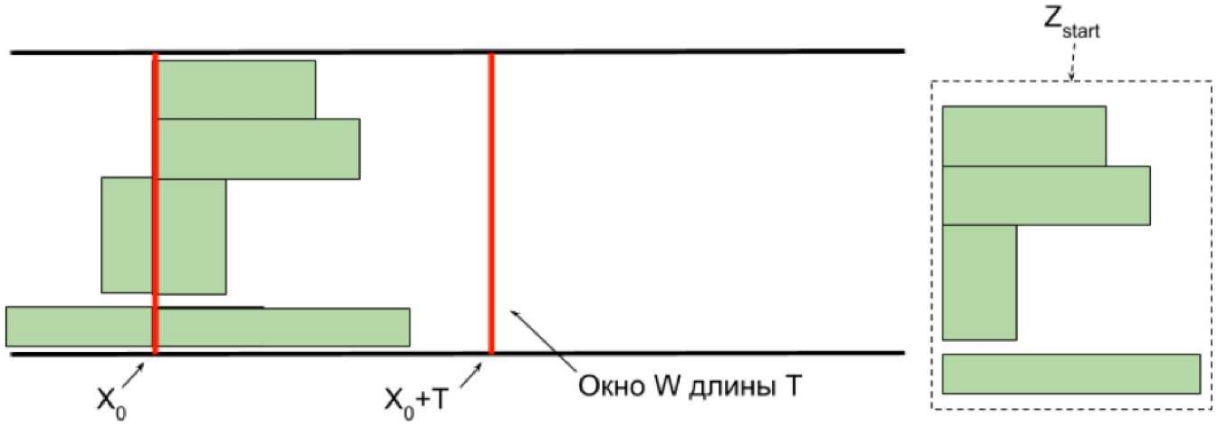

Рис. 4. Множество Z_{start} и пример его положения в окне W

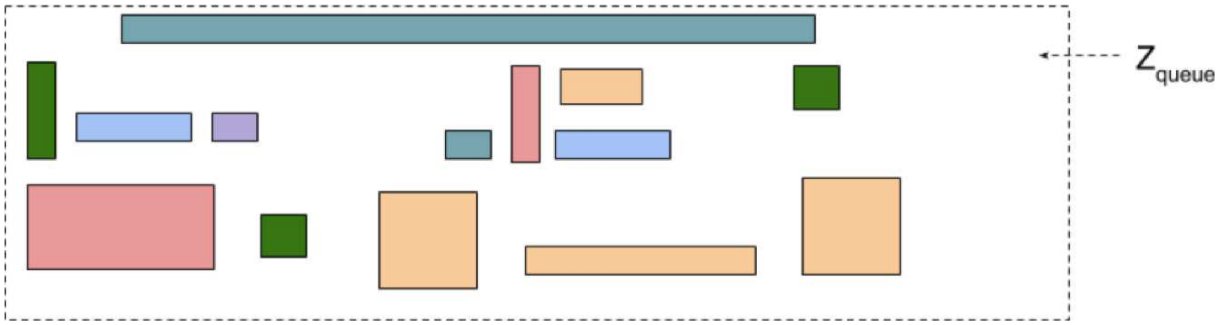

Рис. 5. Множество Z_{queue} заданий, которые находятся в очереди

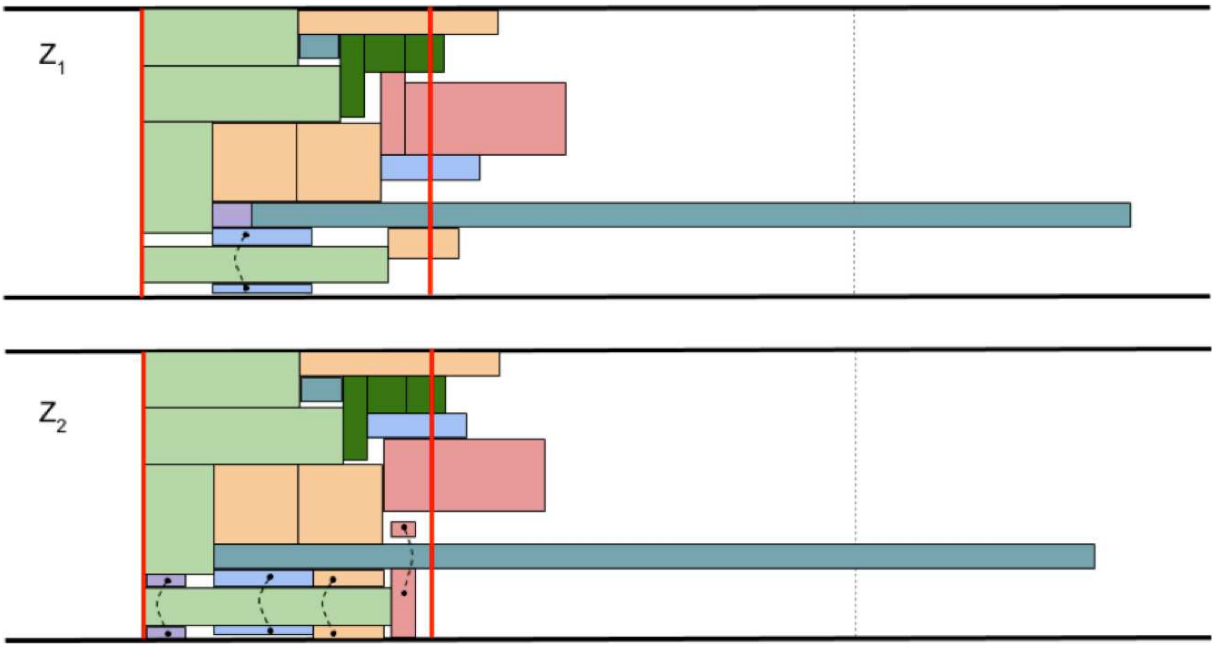

Рис. 6. Примеры двух разных упаковок Z_1 и Z_2

Определим упаковку заданий в окно W как набор Z, который включает в себя все задания из Z_{start} и 0 или более заданий из Z_{queue} с определенными координатами X_i и разбиениями R_i, причем:

— $\forall X_i \in Z_{\text{queue}} \Rightarrow X_0 < X_i \leqslant X_0 + T$;

— $\forall X_i \geqslant \max(X_0, Q_i)$;

— все задания, входящие в упаковку Z, не пересекаются между собой и лежат в полосе.

Поясним свойства упаковки и ограничение, которое накладывается на ее составление. Мы предполагаем, что нельзя изменять составные части заданий из Z_{start} и что в процессе упаковки для любого задания J_i из Z_{queue} разрешается разбивать каждый отдельный прямоугольник из R_i на несколько прямоугольников только таким образом, что:

— размеры по оси Y всех прямоугольников из R_i в сумме дают H_i, т.е. $\sum\limits_j (h_{ij}) = H_i$;

— координата X левой стороны всех прямоугольников равна Xi;

— размер по оси X всех прямоугольников равен Ti;

— не разрешается поворачивать прямоугольники.

Введем функцию потери качества упаковки в виде функции от упаковки Z и параметров окна, обозначим ее $Opt(Z, W)$.

Постановка задачи: на заданном окне и наборах заданий Z_{start} и Z_{queue} найти упаковку заданий Z_{end} в окно W с минимальным значением функции потери качества этой упаковки — $Opt(Z, W)$.

3.2. Характеристики функционирования суперкомпьютера. Пусть $UNum(Z)$ — число пользователей, чьи задачи принадлежат упаковке Z, $UJobs(i)$ — множество задач i-го пользователя в упаковке Z, $Class(A, B)$ — множество задач, для которых H_i принадлежит заданному полуинтервалу $[A, B)$.

Введем способы оценки функции потери качества упаковки Z. Исходя из длительного опыта эксплуатации больших суперкомпьютерных установок, нами были выбраны 5 важнейших характеристик функционирования суперкомпьютерного комплекса, которые являются показателем качества планирования задач на выбранной системе. Для проведения сравнительного анализа все характеристики функционирования суперкомпьютерного комплекса были путем нормировки приведены к одному интервалу $[0, 1]$.

Характеристика: утилизация вычислительных узлов. Смысл: минимизация количества свободных ресурсов (= уменьшаем размер “дырок в упаковке”/нераспределенных ресурсов)

$$Utilization(Z, W) = 1 - \sum_{i=1}^{|Z|} \frac{H_i * \big(\min(T, X_i + T_i\big) - X_i}{H * T}.$$

Характеристика: среднее время старта первого задания пользователей в окне W. Смысл: минимизация среднего расстояния от задачи каждого цвета, у которой $X_i - X_0$ минимально среди задач данного цвета, до начала окна W — X_0

$$FUJST(Z, W) = \sum_{u=1}^{UNum} \min_{j \subset UJobs(u)} \frac{Q_j - X_j}{UNum(Z)}.$$

Характеристика: количество запущенных задач. Смысл: “скорострельность”, минимизация количества не запущенных задач из Z_{queue}

$$StartedJobs(Z, W) = 1 - \frac{|Z_{\text{start}}| + |Z_{\text{queue}}| - |Z|}{|Z_{\text{queue}}|}.$$

Характеристика: количество пользователей, чьи задачи из Z_{queue} были запущены в окне W. Смысл: минимизация количества пользователей, чьи задачи из Z_{queue} не были запущены в окне W

$$StartedUsers(Z, W) = 1 - \frac{UNum(Z_{\text{queue}}) + UNum(Z_{\text{start}}) - UNum(Z)}{UNum(Z_{\text{queue}}) + UNum(Z_{\text{start}})}.$$

Характеристика: среднее время старта задач, принадлежащих определенному классу. Смысл: минимизация среднего расстояния от каждой задачи, чей размер $H_i \in [A, B)$, до начала окна W — X_0

$$AVGST(Z, W, Class) = 1 - \sum_{i \subset Class} \frac{(Q_j - X_j)/T_i}{|Class|}.$$

Изменяющиеся со временем внешние факторы (например, цели использования суперкомпьютерных установок, их пользовательская база, распределение размеров и типов задач этих пользователей) формируют

новую целевую функцию эффективности планирования ресурсов суперкомпьютерного комплекса. Функция выглядит так, как представлено в (1) и (2):

$$\Omega = Utilization, FUJStartTime, StartedJobs, StartedUsers, AVGST, \tag{1}$$

$$S = \sum_{i=1,\ldots,5} (w_i * Characteristics_i), \sum_{i=1,\ldots,5} (w_i) = 1; \quad w \in weights, \quad Characteristics \in \Omega. \tag{2}$$

Получая новую целевую функцию эффективности планирования ресурсов, планировщик на основании информации о текущем потоке задач, истории прохождения потока задач на системе и информации о задачах и пользователях, поставивших их в поток, исторической и текущей архитектуры суперкомпьютерного комплекса, значений характеристик функционирования суперкомпьютерного комплекса за прошедший период создает модель планирования потока задач суперкомпьютерного комплекса при заданной целевой функции и находит на всех возможных наборах настроек такой набор своих настроек, который удовлетворяет максимальному значению формул (1) и (2). Такое значение функции будет оптимальным, а найденный набор параметров планировщика — оптимальным на данном потоке задач. В виду того, что порой задачу поиска оптимального набора параметров планировщика нельзя решить путем перебора (из-за ограничений по времени на данный цикл планирования), то в случае большого потока задач применяются приближенные алгоритмы для поиска максимума функции эффективности (и соответственно параметров планировщика).

Веса характеристик функционирования суперкомпьютеров “Ломоносов”/“Ломоносов-2”/“Mira”

Характеристики	с/к “Ломоносов”	с/к “Ломоносов-2”	с/к “Mira”
Утилизация вычислительных ресурсов	0.5	0.5	0.3
Скорость запуска исполнения первой задачи с момента ее постановки в очередь для каждого пользователя	0.3	0	0
Количество запущенных задач за единицу времени	0	0	0.1
Количество обслуженных пользователей за единицу времени	0	0.3	0.1
Среднее время запуска классов задач	Class “Medium”: 0.2	Class “Medium”: 0.2	Class “Large”: 0.5

4. Практика применения многоцелевого подхода для целевой оптимизации структуры потока задач и сравнения эффективности суперкомпьютерных комплексов. Рассмотрим примеры функций потери качества (целевых функций) для разных целей суперкомпьютерных комплексов. Целям планирования процессорного времени суперкомпьютеров “Ломоносов”, “Ломоносов-2” и “Mira” будут соответствовать веса в нашей базовой формуле (1)–(2), приведенные в таблице. В таблице приведены два класса задач пользователей Class “Medium” и Class “Large”, которые соответствуют средним и большим задачам для упомянутых установок. Для оценки правильности и необходимости применения подхода рассмотрим два реальных потока задач: для суперкомпьютера “Ломоносов” будем рассматривать очередь regular4, а для суперкомпьютера “Ломоносов-2” — compute. Раздел regular4 состоит из 4096 узлов (CPU: Intel Xeon X5570 2.93GHz, объем памяти на GPU: 5.25 GB, модель GPU: Tesla X2070). Раздел compute состоит из 1504 узлов с процессором (CPU: Intel Xeon E5-2697 v3 2.60GHz, CPU-ядер: 14, объем памяти на GPU: 11.56 GB, GPU: Tesla K40s).

На рис. 7 изображены функции эффективности планирования потока задач для потоков задач суперкомпьютеров “Ломоносов” (зеленая линия) и “Ломоносов-2” (желтая линия). Функция эффективности рассчитана по описанной методике (на основе весов, которые представлены в таблице), исходя из целей для суперкомпьютеров “Ломоносов”, “Ломоносов-2” и “Mira” соответственно.

На графике 3 рис. 7, где представлены эффективности суперкомпьютеров “Ломоносов” и “Ломоносов-2”, подсчитанные по формуле для суперкомпьютера “Mira”, можно заметить, что, несмотря на примерно равную утилизацию (утилизация обоих суперкомпьютеров в рассматриваемый период времени была примерно одинакова, что соответствует одинаковой плотности планирования), показатели эффективности разнятся кардинально, показывая, что только утилизация системы не может являться единым определением эффективности для любых крупных вычислительных центров. Необходимо рассматривать каждую отдельную установку, исходя только из целей ее эксплуатации, а не основываясь на общих понятиях, таких как, например, утилизация.

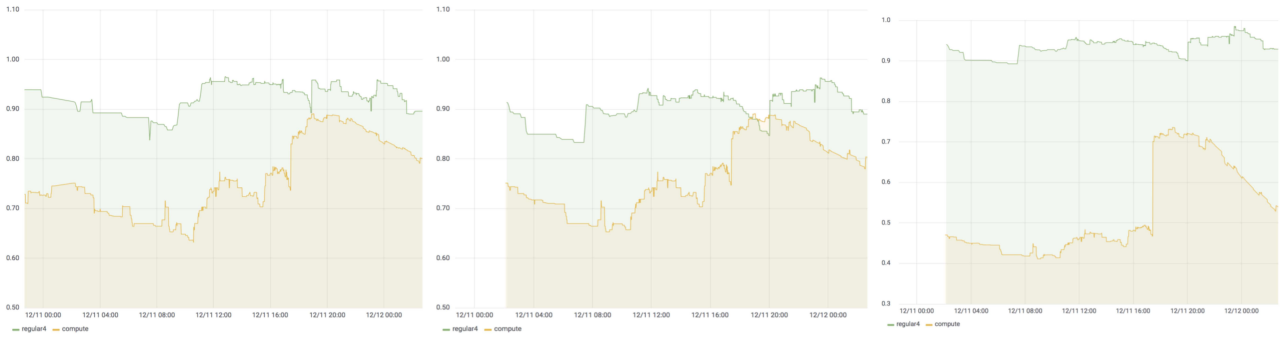

Рис. 7. Сравнение двух потоков задач (“Ломоносов” и “Ломоносов-2”) для всех трех представленных функций эффективности

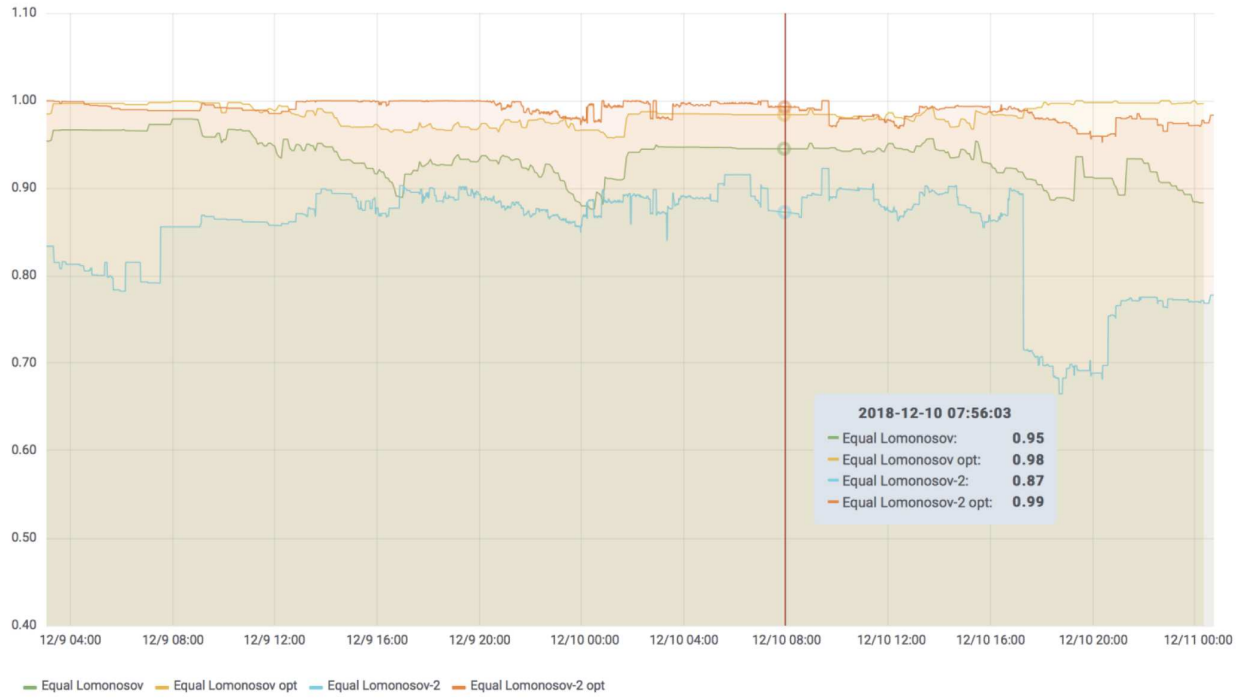

Рис. 8. Сравнение двух потоков задач (“Ломоносов” и “Ломоносов-2”) на примере равновзвешенной функции эффективности и ее возможных оптимальных значений

Другим интересным результатом работы стал оценка оптимально возможной эффективности для данного потока задач и заданной функции эффективности. На рис. 8 приведен пример расчета функции эффективности для разделов regular4 (“Ломоносов”) и compute (“Ломоносов-2”), рассчитанный согласно предложенному подходу со всеми весовыми коэффициентами, равными 0,2. Исходя из данного анализа системный администратор может точнее выбирать настройки планировщика, основываясь на исторических и текущих данных о потоке задач и состоянии суперкомпьютера. Дополнительно благодаря этому подходу администраторы высокопроизводительной системы могут оценивать, насколько хорошо сработали планировщик и менеджеры ресурсов на данном потоке заданий. Иногда проблема кроется не в самом планировщике, а в потоке задач, который можно физически (не параметрами планировщика) изменять исходя из актуального представления о пользователях комплекса.

Для проведения экспериментов и текущего контроля был разработан web-интерфейс для мониторинга состояния эффективности планирования (рис. 9). За основу был взят инструмент Grafana, куда благодаря реализованному серверному решению поставляются данные по текущему состоянию кластера, эффективности и некоторым сверткам отдельных характеристик (рис. 10). Свертки по любым двум характеристикам позволяют исследовать процессы планирования и точнее определять аномалии, которые встречаются в процессе изменения функции эффективности. Системный администратор может в режиме онлайн оценивать текущую и возможную оптимальную эффективность системы. Система апробирована

Рис. 9. Dashboard системы мониторинга состояния потока задач суперкомпьютера

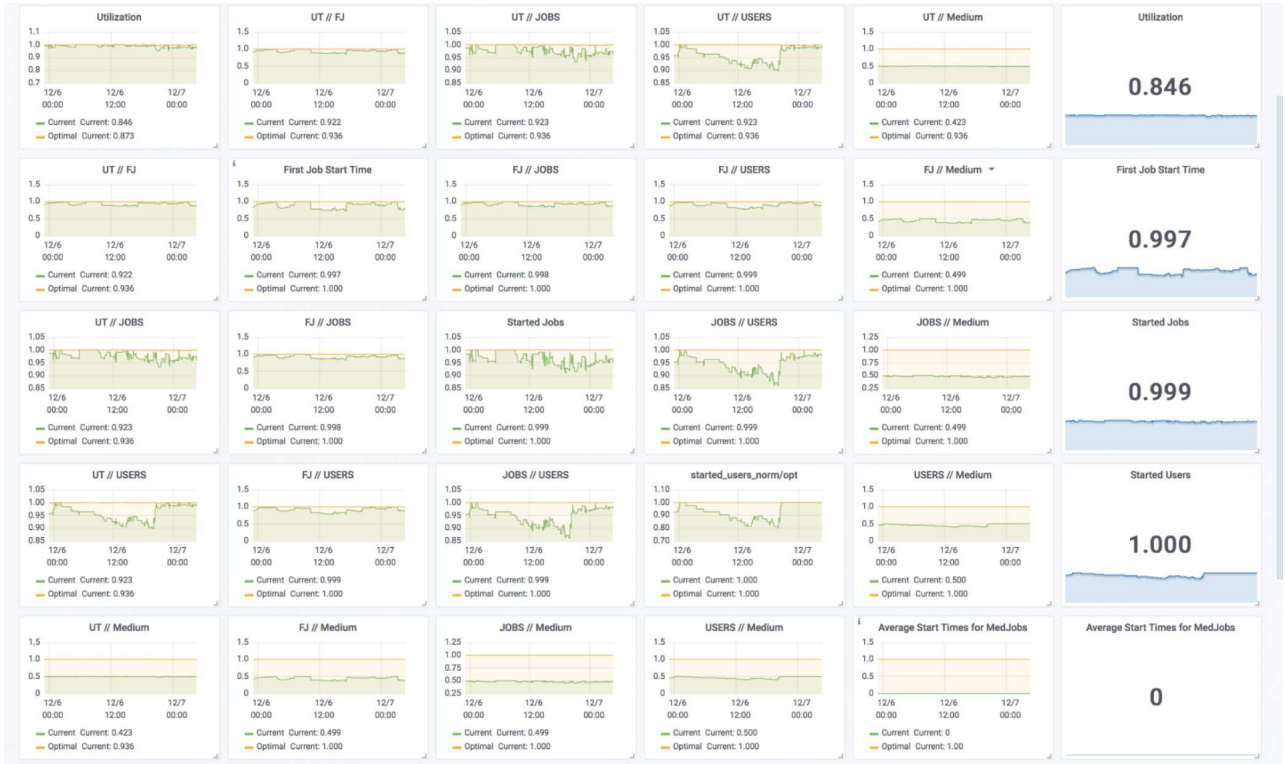

Рис. 10. Свертки отдельных характеристик функционирования суперкомпьютерной системы

на суперкомпьютерах “Ломоносов” и “Ломоносов-2”.

В ближайших планах разработка рекомендательной системы для целевой оптимизации структуры потока задач суперкомпьютерных комплексов, основанная на информации о пользователях и их исторической деятельности на вычислительной системе (размеры задач, использованные пакеты, использование времени, запрошенного для вычислений, работа с сетью, статусы завершения задач и т.д.). Вся собираемая информация позволяет нам предсказывать ближайшие загрузки, параметры задач, делать исходя из этого примеры планирования и выбирать близкие к оптимальным значения настроек планировщика заранее.

5. Заключение. В рамках данной работы был предложен новый подход к оценке эффективности функционирования суперкомпьютерной системы на примере суперкомпьютеров “Ломоносов” и “Ломоносов-2”, основанный на их базовых характеристиках функционирования. Введена функция потери качества планирования для суперкомпьютеров “Ломоносов”, “Ломоносов-2” и “Mira”. Подход позволяет сравнивать разные суперкомпьютерные системы исходя из целей использования, которые стоят перед системными администраторами вычислительных центров. Проведен сравнительный анализ разных функций потери качества планирования на потоках задач рассмотренных суперкомпьютеров. Разработан новый подход к управлению ресурсами суперкомпьютерной системы, основанный на целевой оптимизации функции потери качества предложенного подхода к оценке эффективности. Предложенный подход реализован в рамках единого комплекса управления потоком задач для суперкомпьютерного комплекса, примеры интерфейса приведены в части 4 данной работы. Результаты работы апробированы на системах “Ломоносов” и “Ломоносов-2”.

Работа выполнена с использованием оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ им. М.В. Ломоносова. Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18–29–03230мк.

СПИСОК ЛИТЕРАТУРЫ

1. *Воеводин Вл., Жуматий С., Соболев С., Антонов А., Брызгалов П., Никитенко Д., Стефанов К., Воеводин Вад.* Практика суперкомпьютера “Ломоносов” // Открытые системы. 2012. № 7. 36–39.
2. *Аветисян А.И., Грушин Д.А., Рыжов А.Г.* Системы управления кластерами // Труды ИСП РАН. 2002. **3**. 39–62.
3. *Staples G.* TORQUE resource manager // Proceedings of the 2006 ACM/IEEE Conference on Supercomputing. New York: ACM Press, 2006. doi 10.1145/1188455.1188464.
4. *Klusáček D., Chlumský V., Rudová H.* Optimizing user oriented job scheduling within TORQUE. http://sc13.supercomputing.org/sites/default/files/PostersArchive/tech_posters/post185s2-file3.pdf.
5. *Chlumský V., Klusáček D., Ruda M.* Planning, predictability and optimization within the TORQUE scheduler // MEMICS 2012. Brno: Novpress, 2012. 96–97.
6. Torque Resource Manager. https://www.adaptivecomputing.com/products/torque/.
7. Loadleveler, ВМК МГУ. http://hpc.cmc.msu.ru/bgp/jobs/loadleveler.
8. Documentation Update: IBM LoadLeveler Version 5 Release 1. https://www.ibm.com/support/knowledgecenter/SSFJTW_5.1.0/loadl51_content.html.
9. IBM LoadLeveler to IBM Platform LSF Migration Guide, An IBM Redpaper publication. http://www.redbooks.ibm.com/redpapers/pdfs/redp5048.pdf
10. *Жуматий С.А.* Система управления заданиями Cleo. Руководство пользователя, 2007. http://www.hpc.icc.ru/documentation/cleo_ug.pdf.
11. Parallel Job Control System. http://suppz.jscc.ru.
12. *Баранов А.В., Тихомиров А.И.* Планирование заданий в территориально распределенной системе с абсолютными приоритетами // Вычислительные технологии. 2017. **22**, Спец. вып. 1. 4–12.
13. MAUI Cluster Scheduler. https://ru.wikipedia.org/wiki/Maui_(программа).
14. Maui Scheduler Administrator’s Guide. URL:http://docs.adaptivecomputing.com/maui/.
15. Moab HPC Suite. http://www.adaptivecomputing.com/moab-hpc-basic-edition/.
16. *Jackson D., Snell Q., Clement M.* Core algorithms of the Maui scheduler // Lecture Notes in Computer Science. Vol. 2221. Heidelberg: Springer, 2001. 87–102.
17. Comparison of cluster software. https://en.wikipedia.org/wiki/Comparison_of_cluster_software.
18. *Yoo A.B., Jette M.A., Grondona M.* SLURM: Simple Linux Utility for Resource Management // Lecture Notes in Computer Science. Vol. 2862. Heidelberg: Springer, 2003. 44–60.
19. *Novotny M.* Job scheduling with the SLURM resource manager. https://is.muni.cz/th/173052/fi_b_b1/thesis.pdf.
20. *Lipari D.* The SLURM Scheduler Design // SLURM User Group. http://slurm.schedmd.com/slurm_ug_2012/SUG-2012-Scheduling.pdf.

21. *Леоненков С.Н., Жуматий С.А.* Алгоритмы планирования и эффективность использования суперкомпьютера “Ломоносов” // В сб. “Вычислительные технологии в естественных науках. Методы суперкомпьютерного моделирования”. Серия Механика, управление и информатика. Т. 4. М.: ИКИ РАН, 2017. 53–63.
22. *Jones M.* Optimization of resource management using supercomputers SLURM. 2012. http://www.ibm.com/developerworks/ru/library/l-slurm-utility/.
23. Argonne Leadership Computing Facility. https://www.alcf.anl.gov.
24. *Allcock W., Rich P., Fan Y., Lan Z.* Experience and practice of batch scheduling on Leadership Supercomputers at Argonne. http://jsspp.org/papers17/paper_19.pdf.
25. SLURM User Group Meeting. https://sc18.supercomputing.org/proceedings/bof/bof_pages/bof106.html.
26. PBS Professional Open Source Project. https://www.pbspro.org.
27. Altair PBS Professional: Overview. https://www.pbsworks.com/PBSProduct.aspx?n=Altair-PBS-Professional&c=Overview-and-Capabilities.
28. LoadLeveler. IBM Knowledge Center. https://www.ibm.com/support/knowledgecenter/en/SSFJTW/loadl_welcome.html.

Поступила в редакцию
22.04.2019

Target Optimization of a Supercomputer Task Flow

S. N. Leonenkov [1]

[1] *Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics; Leninskie Gory, Moscow, 119992, Russia; Graduate Student, e-mail: leonenkov@cs.msu.ru*

Received April 22, 2019

Abstract: This paper is a result of studying the task flows observed on the Lomonosov and Lomonosov-2 supercomputers. A new approach to evaluating the performance of a supercomputer system based on its basic performance characteristics is proposed. A supercomputer's scheduling efficiency function is introduced for Lomonosov, Lomonosov-2 and other systems. The approach allows the system administrators to compare various supercomputer systems based on their usage aims. This paper describes the Moscow State University experience of applying the proposed approach to the optimization of Lomonosov and Lomonosov-2 scheduling resources.

Keywords: supercomputer, scheduling efficiency, scheduling algorithms, SLURM.

References

1. Vl. V. Voevodin, S. A. Zhumatii, S. I. Sobolev, et al., “The Lomonosov Supercomputer in Practice,” Otkrytye Sistemy, No. 7, 36–39 (2012).

2. A. I. Avetisyan, D. A. Grushin, and A. G. Ryzhov, “Cluster Control Systems,” Tr. Mat. Inst. Sistemnogo Programm. Ross. Akad. Nauk **3**, 39–62 (2002).

3. G. Staples, “Torque Resource Manager,” in *Proc. 2006 ACM/IEEE Conference on Supercomputing, Tampa, USA, November 11–17, 2006* (ACM Press, New York, 2006), doi 10.1145/1188455.1188464

4. D. Klusácek, V. Chlumský, and H. Rudová, “Optimizing User Oriented Job Scheduling within TORQUE,” in *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis, Denver, USA, November 17–21, 2013.*
http://sc13.supercomputing.org/sites/default/files/PostersArchive/tech_posters/post185s2-file3.pdf.
Cited May 28, 2019.

5. V. Chlumský, D. Klusácek, and M. Ruda, “Planning, Predictability and Optimization within the TORQUE Scheduler,” in *MEMICS 2012* (Novpress, Brno, 2012), pp. 96–97.

6. TORQUE Resource Manager. https://www.adaptivecomputing.com/products/torque/. Cited May 28, 2019.

7. LoadLeveler. http://hpc.cmc.msu.ru/bgp/jobs/loadleveler. Cited May 28, 2019.

8. Documentation Update: IBM LoadLeveler Version 5 Release 1.
https://www.ibm.com/support/knowledgecenter/SSFJTW_5.1.0/loadl51_content.html. Cited May 28, 2019.

9. IBM LoadLeveler to IBM Platform LSF Migration Guide, An IBM Redpaper publication. http://www.redbooks.ibm.com/redpapers/pdfs/redp5048.pdf. Cited May 28, 2019.

10. S. A. Zhumatii, *Job Control System Manual*. http://www.hpc.icc.ru/documentation/cleo_ug.pdf. Cited May 28, 2019.

11. Parallel Job Control System. http://suppz.jscc.ru. Cited May 28, 2019.

12. A. V. Baranov and A. I. Tikhomirov, "Scheduling of Jobs in a Territorially Distributed Computing System with Absolute Priorities," Vychisl. Tekhnol. **22** (Suppl. 1), 4–12 (2017).

13. Maui Cluster Scheduler. https://ru.wikipedia.org/wiki/Maui_(программа).

14. Maui Scheduler Administrator's Guide. http://docs.adaptivecomputing.com/maui/. Cited May 28, 2019.

15. Moab HPC Suite. http://www.adaptivecomputing.com/moab-hpc-basic-edition/. Cited May 28, 2019.

16. D. Jackson, Q. Snell, and M. Clement, "Core Algorithms of the Maui Scheduler," in *Lecture Notes in Computer Science* (Springer, Heidelberg, 2001), Vol. 2221, pp. 87–102.

17. Comparison of Cluster Software. https://en.wikipedia.org/wiki/Comparison_of_cluster_software. Cited May 28, 2019.

18. A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple Linux Utility for Resource Management," in *Lecture Notes in Computer Science* (Springer, Heidelberg, 2003), Vol. 2862, pp. 44–60.

19. M. Novotny, *Job Scheduling with the SLURM Resource Manager* (Masarykova Univ., Bachelor Thesis, Brno, 2009).

20. D. Lipari, "The SLURM Scheduler Design," http://slurm.schedmd.com/slurm_ug_2012/SUG-2012-Scheduling.pdf.

21. S. N. Leonenkov and S. A. Zhumatiy, "Scheduling Algorithms and Efficiency of Lomonosov Supercomputer," in *Computing Technologies in Natural Science* (Inst. Kosmich. Issled. Ross. Akad. Nauk, Moscow, 2017), Vol. 4, 53–63.

22. M. Jones, "Optimization of Resource Management Using Supercomputers SLURM," http://www.ibm.com/developerworks/ru/library/l-slurm-utility. Cited May 28, 2019.

23. Argonne Leadership Computing Facility. https://www.alcf.anl.gov

24. W. Allcock, P. Rich, Y. Fan, and Z. Lan, "Experience and Practice of Batch Scheduling on Leadership Supercomputers at Argonne," http://jsspp.org/papers17/paper_19.pdf. Cited May 28, 2019.

25. SLURM User Group Meeting. https://sc18.supercomputing.org/proceedings/bof/bof_pages/bof106.html. Cited May 28, 2019.

26. PBS Professional Open Source Project. https://www.pbspro.org. Cited May 28, 2019.

27. Altair PBS Professional: Overview. https://www.pbsworks.com/PBSProduct.aspx?n=Altair-PBS-Professional&c=Overview-and-Capabilities. Cited May 28, 2019.

28. LoadLeveler. IBM Knowledge Center. https://www.ibm.com/support/knowledgecenter/en/SSFJTW/loadl_welcome.html. Cited May 28, 2019.