

DOI: 10.34020/2073-6495-2021-3-019-030

УДК 004.75

## **ОСОБЕННОСТИ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ЗАДАНИЙ В ТЕРРИТОРИАЛЬНО-РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ**

**Агапов Е.В., Бобров Л.К., Зайков К.А.**

Новосибирский государственный университет  
экономики и управления «НИНХ»

E-mail: agapovefim@gmail.com, l.k.bobrov@edu.nsuem.ru

Работа посвящена рассмотрению основных распространенных алгоритмов планирования заданий в территориально-распределенных вычислительных системах. Характеризуются специфические особенности алгоритмов и приводится их сравнительный анализ в соответствии с выбранными критериями. Определяются основные факторы, которые должны быть учтены при построении алгоритмов управления заданиями в территориально-распределенных вычислительных системах.

*Ключевые слова:* территориально-распределенные вычислительные системы, архитектура, управление заданиями, алгоритмы, сравнительный анализ.

## **SPECIFICITY OF TASK SCHEDULING ALGORITHMS IN GEOGRAPHICALLY DISTRIBUTED COMPUTER SYSTEMS**

**Agapov E.V., Bobrov L.K., Zaykov K.A.**

Novosibirsk State University of Economics and Management

E-mail: agapovefim@gmail.com, l.k.bobrov@edu.nsuem.ru

The work is devoted to the consideration of the main widespread algorithms for scheduling tasks in geographically distributed computing systems. The specific features of the algorithms are characterized and their comparative analysis is presented in accordance with the selected criteria. The main factors that should be taken into account when constructing job control algorithms in geographically distributed computing systems are determined.

*Keywords:* geographically distributed computing systems, architecture, job control task, algorithms, comparative analysis.

### **ВВЕДЕНИЕ**

Актуальным направлением в области суперкомпьютерных технологий является повышение производительности вычислительных комплексов (ВК) путем объединения их в единую территориально-распределенную вычислительную систему (ТРВС). Подобное объединение ресурсов отвечает направлению развития Национального проекта «Наука», который ориентирован на внедрение передовых технологий для проведения научно-исследовательских работ [32]. Также данное направление актуально для развития новых цифровых технологий, которые определены в национальной программе «Цифровая экономика» [33]. В материалах заседания Президиума Совета при Президенте Российской Федерации по стратегическому

развитию и национальным проектам говорится о том, что для достижения поставленной цели Национальных проектов необходимо эффективное партнерское взаимодействие государства, исследовательских институтов и высокотехнологичных компаний [31], куда в настоящее время относят центры коллективного пользования (ЦКП). Основной задачей ЦКП является предоставление вычислительной среды сторонним организациям для проведения расчетов. Для организации доступа к ресурсам ЦКП разрабатываются различные решения, позволяющие упростить процесс предоставления услуг и повысить качество обслуживания.

Одним из важных направлений в области распределенных вычислений является планирование заданий, которое известно научному сообществу со второй половины прошлого века. К числу первых наиболее значимых работ относят исследование Ричарда Конвея [10], где обосновывается заключение о том, что алгоритмы, используемые при планировании заданий, относятся к классу NP (non-deterministic polynomial), и это говорит о их сложности. Поэтому рассматриваемое направление имеет огромный интерес у научного сообщества, и авторы публикуют многочисленные теоретические и практические результаты своих работ [1, 6–9, 11–19, 21–24, 27, 28, 30, 34]. В данной статье выделяются перспективные направления развития алгоритмов планирования заданий и проводится их сравнительный анализ.

### **НАПРАВЛЕНИЯ РАЗВИТИЯ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ЗАДАНИЙ**

Задача объединения вычислительных комплексов в единую систему приобрела актуальность с 1970-х гг., когда Сеймур Крей совершал попытки создания первого суперкомпьютера «Cray-1». Одним из ключевых моментов стала разработка им комплекса «CDC 8600», который объединял в одну систему четыре «CDC 7600». В этих условиях перед специалистами остро встала задача поиска алгоритмов планирования заданий, обеспечивающих повышение эффективности системы в целом. Сложность задачи сильно зависела от приоритетов вычислительных заданий, топологии комплекса, количества параллельных процессоров задания, времени обработки задания и выбранных критериев. Поэтому задача планирования заданий являлась вычислительно трудноразрешимой даже при упрощенных предположениях [25].

В работах [8, 14, 15, 18, 19] авторами были предложены различные эвристические алгоритмы решения задачи планирования. Кашара и Нарита [18, 19] предложили эвристический алгоритм критического пути и алгоритм приближения (поиск в глубину).

В работе [8] был предложен алгоритм поиска в пространстве состояний. Хеллстрем и Кана [15] предложили асимметричную модель нейронной сети, позволяющей формировать очередь из заданий. Предлагая свой алгоритм, авторы указывали, что базовая архитектура суперкомпьютера состояла из однотипных узлов, имеющих одинаковые системные характеристики и используемое ПО. По этой причине рассмотренные алгоритмы показывают хороший результат в локальной системе управления заданиями в пределах одного комплекса, но при организации глобальной очереди, где следует вы-

брать ВК в зависимости от системных и программных требований задания, результат будет отличаться от ожидаемого.

С учетом гетерогенности реальных систем в работах [6, 7, 9, 11] авторами предлагаются подходы к управлению потоком заданий, основанные на прогнозе будущей производительности каждого узла. Хотя они демонстрируют хороший результат для прогнозирования вычислительной нагрузки центрального процессора, в основе их подхода лежит исторический анализ загруженности процессоров, и они трудно реализуемы на крупных распределенных вычислительных системах из-за роста сложности при масштабировании системы и, соответственно, длительного времени поиска решений.

Для повышения производительности и поиска приемлемых решений предлагаются различные метаэвристические алгоритмы, которые сочетают в себе информацию о работе системы и прогнозируемые данные, что позволяет регулировать время поиска решений путем выбора количества итераций алгоритма. Существуют следующие популярные метаэвристические подходы: метод роя частиц [30], алгоритм имитации отжига [21], муравьиный алгоритм [22], генетический алгоритм [27] и многоцелевые эволюционные алгоритмы [16].

В основе каждого подхода заложена идея о взаимодействии децентрализованных взаимосвязанных элементов, образующих самоорганизующуюся систему. В данном случае систему образует множество программных агентов, каждый из которых ориентирован на решение определенной функциональной задачи. Агенты взаимодействуют друг с другом и внешней средой, в совокупности обеспечивая при необходимости решение задач повышенной сложности для достижения поставленных общесистемных целей.

Муравьиный алгоритм получил свое название из-за реализуемых в нем математических моделей, основанных на процессе поиска агентами кратчайших путей [22]. При поиске маршрутов агенты помечают пути своего движения, и чем больше агентов выбирают один и тот же маршрут, тем больше агентов может его использовать. Движение по короткому пути занимает меньше времени, поэтому выбравшие его агенты успевают пройти по нему большее число раз, и это дает основание для завершения процесса поиска наиболее короткого пути. Алгоритм сводится к многократному обходу некоторого графа, дуги которого имеют не только веса, но и динамически меняющиеся характеристики. Тем не менее в данном случае правила поведения агентов достаточно просты для формального математического описания.

В основе метода роя частиц лежит идея перемещения частиц в пространстве решений. Каждая частица «запоминает» наилучшую точку в пространстве решений, в которой была, и стремится в нее вернуться. Однако ее движение подвержено силе инерции, и это вызывает некоторые стохастические изменения траектории движения. Рой имеет общую память, поэтому каждая частица знает координаты наилучшей точки среди всех, в которых была любая частица роя. То есть наилучшее решение, найденное роем в каждый момент времени, известно всем его частицам. В итоге на движение частицы влияют стремление к своему наилучшему положению, стремление к наилучшему среди всех частиц положению, инерционность и случайные отклонения.

Важным свойством рассмотренных алгоритмов является зависимость их эффективности от используемых в них коэффициентов. Поскольку они имеют возможность принимать неограниченное количество значений из некоторого диапазона, появляется необходимость в нахождении способа их подбора. На практике для этого применяется генетический алгоритм, который является одной из разновидностей многоцелевого эволюционного алгоритма. Граф с последовательностью выполнения заданий корректируется по мере выполнения генетического алгоритма. Генетический алгоритм подразумевает наличие заданного количества итераций, на каждой из которых оценивается качество каждой из схем планирования. Одним из критериев эффективности является уменьшение времени работы ТРВС в целом. Лучшие из схем планирования выбираются для сочетания между собой и создания новых на их основе. В качестве критерия останова может выступать такой параметр, как отсутствие прироста быстродействия или его ухудшение. После завершения работы выбирается наиболее эффективная схема планирования.

Как утверждают авторы работ [13, 24], одним из ключевых недостатков генетических алгоритмов является то, что абстрактную структуру генетического алгоритма трудно реализовать эффективно в контексте пространственных многокритериальных систем. Альтернативой генетическим алгоритмам в работах [12, 17, 23] рассмотрен эволюционный алгоритм с негенетическим локальным поиском путей улучшения генотипа (меметический алгоритм). Алгоритм был апробирован на статическом наборе заданий, где время обработки заданного количества итераций сократилось в 2 раза. Но статичность набора заданий подразумевает, что количество поступивших заданий не изменялось во времени. Обслуживание потока заданий в ТРВС принципиально отличается от обработки статичных наборов заданий, в данном случае задания поступают в случайные моменты времени, их параметры также случайны, следовательно, детерминированный выбор ВК для постановки заданий на расчет исключен.

В настоящее время на практике чаще всего реализуют следующие алгоритмы для планирования заданий:

– FCFS (First Come First Serve): задание, поставленное в очередь раньше остальных, будет иметь наивысший приоритет и рассчитываться в первую очередь [28];

– SJF (Shortest-Job-First): задание, для выполнения которого требуется меньше всего процессорного времени, будет иметь наивысший приоритет; алгоритм LJF (Longest Job First) ориентирован на противоположное условие [28];

– RR (Round Robin): задания распределяются в системе по круговому циклу, при использовании внутри небольших ВК этот алгоритм реализует тот же принцип, что и FCFS.

На их основе разрабатываются более сложные алгоритмы. Например, широко используемым (применяется на 60 % суперЭВМ из Top50 [34], а также в системах управления заданиями «Slurm» и «Torque») и наиболее эффективным является алгоритм Backfill (Обратного заполнения), основанный на механизме FCFS. Ключевая особенность данного алгоритма заключается в возможности переноса заданий на более раннее время, если это не

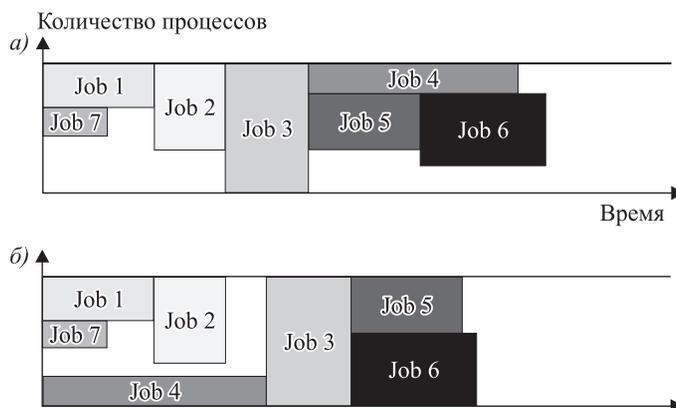


Рис. 1. Алгоритмы (а) conservative backfilling и (б) easy backfilling [1]

повлечет задержки выполнения других заданий в очереди. Выделяют два основных подвида алгоритма Backfill [1, 3]: консервативный (Conservative backfilling) и простой (Easy backfilling).

Консервативный (Conservative backfilling): допустимы переносы заданий вперед, если это не приводит к задержкам выполнения других заданий в очереди. Если алгоритм комбинируется с FCFS, то обеспечивается «справедливое» распределение ресурсов и появляется возможность точного прогноза времени отклика. Однако данный алгоритм не гарантирует оптимального распределения ресурсов, так как количество доступных ресурсов в отдельный момент времени может быть недостаточно для переноса на более ранний срок заданий из очереди (рис. 1, а).

Простой (Easy backfilling): допустимы переносы заданий вперед, если это не приводит к задержке выполнения первой ожидающей задачи. Данный алгоритм обеспечивает более эффективное использование ресурсов при высокой загрузке, но часто отдает предпочтение наименее ресурсоемким заданиям, что может приводить к задержке остальных заданий. Так как задания часто откладываются, возможность точного прогнозирования времени отклика отсутствует (рис. 1, б).

### СТРУКТУРА ТРВС

При проведении анализа алгоритмов следует учитывать структурные особенности ТРВС при организации вычислительного процесса. Зачастую системы подобного рода строятся в соответствии с иерархическим принципом [4], где первым уровнем является ВК (рис. 2). Необходимо отметить важную особенность данного уровня, которая заключается в том, что ВК является самостоятельной, полноценной и независимой системой, имеющей в своем составе механизмы и программы для обработки пользовательских заданий. Вторым уровнем следует считать объединение ВК локальной сетью в пределах одной организации. Верхний уровень ТРВС – это уровень распределенной сети, объединяющий разные территориально удаленные организации.

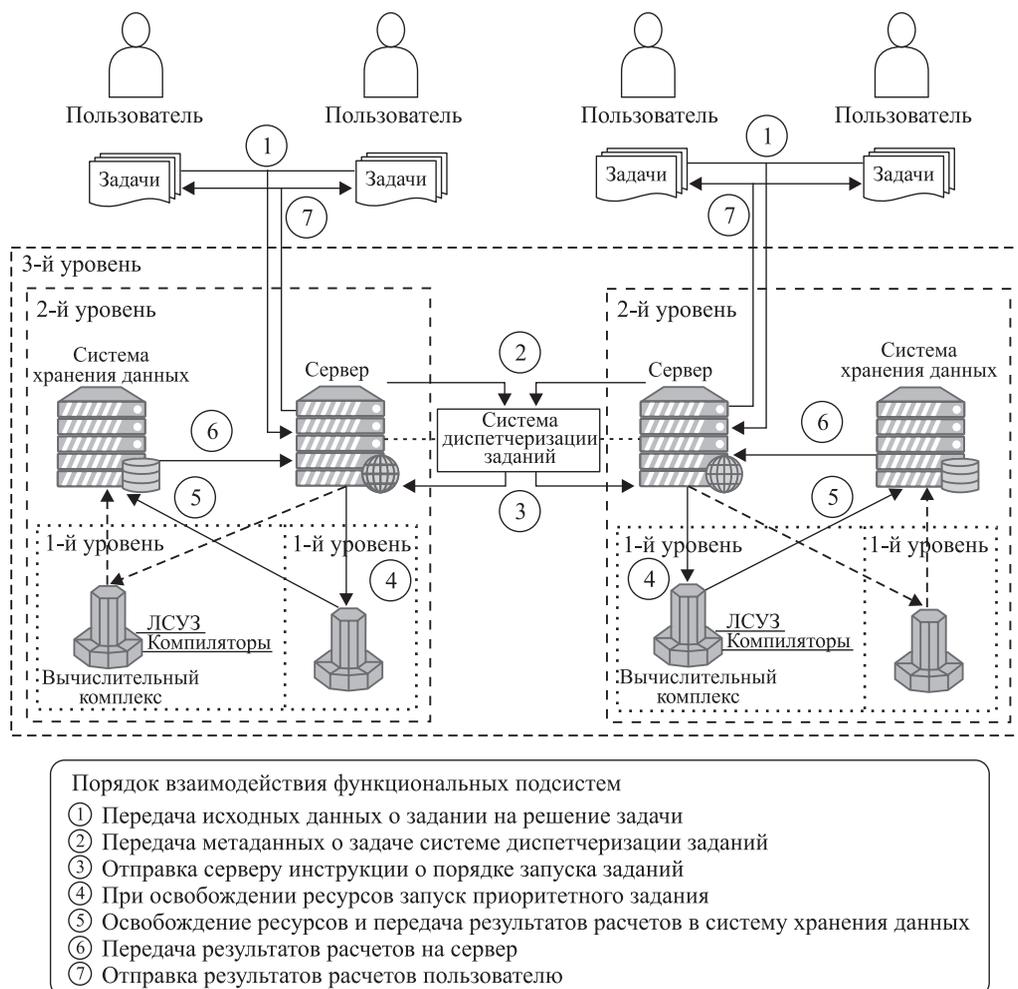


Рис. 2. Структурная схема организации ТРВС

Одной из ключевых подсистем ТРВС следует считать систему диспетчеризации заданий, которая предоставляет механизмы организации глобальной очереди и, как следствие, перераспределения вычислительной нагрузки на основе заданных алгоритмов, что повышает эффективность использования ресурсов. В случае недоступности, неисправности или перегруженности ВК задания пользователей перенаправляются на ресурсы других ВК, что сокращает время ожидания в очереди и повышает качество обслуживания пользователей как потребителей услуг.

### СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ЗАДАНИЙ

Принимая во внимание современные тенденции развития суперкомпьютерных технологий [5] и структурную реализацию ТРВС [4], в таблице представлен сравнительный анализ алгоритмов по следующим критериям, приведенным ниже.

**Алгоритмы планирования заданий**

№	Алгоритм	Год и авторы	Номер критерия сравнения							Комментарии
			1	2	3	4	5	6	7	
1	Алгоритм критического пути	1984 H. Kasahara, S. Narita [18, 19]	+	+	-	+	+	+	-	Ключевым недостатком является то, что раннее завершение задания не повлияет на перераспределение очереди, что приведет к простоям ресурсов ТРВС
2	Алгоритм приближения		+	-	+	+	+	+	-	
3	Алгоритм поиска в пространстве состояний	1988 C. Chen, C. Lee, E. Hou [8]	+	+	-	+	+	+	-	В локальном максимуме стратегия поиска экстремума останавливается, что не позволяет определить наилучший результат. Имеется шанс заклинивания алгоритма
4	Метод роя частиц	1995 J. Kennedy, R. Eberhart [30]	+	+	+	+	-	+	-	Сильная зависимость качества сходимости от выбранных коэффициентов
5	Алгоритм имитации отжига	1983 S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi [21]	+	+	+	+	-	+	-	При изменении состояния ТРВС необходимо экспериментальным путем подбирать новые параметры
6	Муравьиный алгоритм	1992 A. Colomi [22]	+	+	+	+	-	+	-	
7	Генетический алгоритм	1975 J.H. Holland [27]	+	+	+	+	-	-	-	Начальные параметры расчетов влияют на быстрдействие алгоритма и точность расчетов
8	Многоцелевой эволюционный алгоритм	1986 J.D. Farmer, N.H. Packard, A.S. Perelson [16]	+	+	+	+	-	-	-	Производительность алгоритма зависит от его начальных параметров
9	Меметический алгоритм	1992 P. Moscato, M. Norman [12]	+	+	+	+	-	+	-	Нет способа рассчитать критерии окончания работы алгоритма. При изменении состояния ТРВС необходимо экспериментальным путем подбирать новые параметры
10	Асимметричная модель нейронных сетей	1992 B. Hellstrom, L. Kanal [15]	+	+	-	+	-	+	+	При изменении состояния ТРВС необходимо переобучение нейронной сети
11	Easy backfilling	1995 [1]	+	-	+	+	+	+	-	Алгоритм не способен справедливо распределить задания в случае существования конкуренции за вычислительные ресурсы

1. Наличие нескольких уровней приоритетов вычислительных заданий с целью обеспечения качественного планирования [5].

2. Установление очередности запуска программ в случае существования конкуренции за вычислительные ресурсы [5].

3. Обеспечение надежности и отказоустойчивости, а именно – при выходе из строя одного ВК, система должна продолжить распределение заданий в штатном режиме [4].

4. Использование единой системы хранения данных с целью исключения потери информации и обеспечения возможности, в случае необходимости, запускать задание на других ВК [4].

5. Совершенствование ТРВС должно предусматривать постоянное увеличение числа одновременно решаемых пользовательских заданий, повышение их сложности, расширение числа и типов, включенных в систему ВК, повышение производительности [4].

6. Вычислительная сложность алгоритма не должна влиять на производительность системы [5].

7. При планировании заданий должны учитываться такие факторы, как состояние ВК и исходные данные задания, которые со временем подвергаются технологическим изменениям. Так, например, с развитием суперкомпьютерных технологий изменяются требования пользовательских заданий к вычислительным ресурсам, что приводит к росту числа платформозависимых заданий. Компиляторы оптимизируют и адаптируют программу под тот ВК, на котором установлены. Даже незначительная гетерогенность в ВК не позволит перенаправить исполняемые пользовательские программы между ВК, что, как следствие, усложняет алгоритмы распределения заданий.

Во многих работах, касающихся разработок алгоритмов управления заданиями (см., напр., работы [2, 10, 26, 29]), принято сравнивать эффективность предлагаемого алгоритма по отношению к алгоритму Easy backfilling из-за простоты его реализации и широкого распространения на практике.

Как следует из приведенной таблицы, к недостаткам рассмотренных алгоритмов относятся:

1) снижение качества работы алгоритма при масштабировании ТРВС, поскольку при добавлении новых ВК в ТРВС часть алгоритмов требует перерасчета начальных параметров;

2) наличие жесткой привязки задания к ВК.

Принимая это во внимание, следует отметить и некоторые моменты субъективного характера, касающиеся управления заданиями. Например, по завершении отладочных работ над заданием пользователь в силу определенных соображений может и не ставить задание на расчет, а запланировать его запуск при полной загруженности ВК, в то время как другие ВК простаивают. Исходя из этого, в алгоритме планирования заданий также должен учитываться «человеческий фактор».

## ЗАКЛЮЧЕНИЕ

В данной работе представлен краткий обзор направлений развития алгоритмов планирования заданий в ТРВС. Результаты проведенного анализа позволяют сделать вывод о сложности построения алгоритма планирова-

ния заданий в ТРВС, который позволял бы учитывать трудноформализуемые данные о системе и наличие субъективных требований пользователей.

В то же время нивелировать сложность учета этих факторов возможно, например, используя методы нечеткой логики и машинного обучения, а также инструментарий теории очередей и нечетких множеств [7, 9, 11, 30]. Однако это требует дальнейших исследований и поиска соответствующих алгоритмов.

### Литература

1. Мазалов В.В., Никитина Н.Н. Оценка характеристик алгоритма Backfill при управлении потоками задач на вычислительном кластере // Вычислительные технологии. 2012. Т. 17. № 5. С. 71–79.
2. Мамойленко С.Н., Ефимов А.В. Алгоритмы планирования решения масштабируемых задач на распределенных вычислительных системах // Вестник ГОУ ВПО «СибГУТИ». 2010. № 2. С. 66–78.
3. Волович К.И. Методы и алгоритмы организации вычислительного процесса в гибридном высокопроизводительном комплексе на основе виртуальной среды исполнения: дис. ... канд. техн. наук: 05.13.15. М., 2019. 114 с.
4. Тихомиров А.И. Методы и средства организации системы управления вычислительными заданиями в территориально распределенной сети суперкомпьютерных центров коллективного пользования: дис. ... канд. техн. наук: 05.13.11. М., 2019. 143 с.
5. Тютляева Е.О., Одинцов И.О., Московский А.А., Мармузов Г.В. Тенденции развития вычислительных узлов современных суперкомпьютеров // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». 2019. С. 92–114.
6. Carrington L.A., Snavelly A., Wolter N. Performance Prediction Framework for Scientific Applications // Future Generation Computer Systems. 2006. Vol. 22. P. 336–346.
7. Cirne W., Berman F. A Comprehensive Model of the Supercomputer Workload // Proc. IEEE Fourth Ann. Workshop Workloads Characterization. 2001. P. 140–148.
8. Chen C.L., Lee C.S., Hou E.S. Efficient scheduling algorithms for robot inverse dynamics computation on a multiprocessor system // IEEE Trans. System Man Cybernetics. 1988. Vol. 18. P. 729–743.
9. Clement M.J., Quinn M.J. Analytical Performance Prediction on Multicomputers // J. Supercomputing. 1993. P. 886–894.
10. Conway R.W., Maxwell W., Miller L. Theory of Scheduling // Massachusetts Addison Wesley Publications. 1967. P. 304.
11. Dinda P.A., O'Hallaron D.R. Host Load Prediction Using Linear Models // Cluster Computing. 2000. Vol. 3. P. 265–280.
12. Feitelson D., Rudolph L.H., Schwiegelshohn U., Wong P. Theory and Practice in Parallel Job Scheduling // Theoretical Computer Science. 2004. P. 17–47.
13. Gandhi T., Nitin, Alam T. Quantum Genetic Algorithm with Rotation Angle Refinement for Dependent Task Scheduling on Distributed Systems // Tenth International Conference on Contemporary Computing (IC3). 2017. P. 12–15.
14. Gonzalez M.J. Deterministic processor scheduling // Computing Surveys. 1977. Vol. 9. № 3. P. 173–204.
15. Hellstrom B., Kanal L. Asymmetric mean-field neural networks for multiprocessor scheduling // Neural Networks. 1992. Vol. 5. P. 671–686.
16. Hisao I., Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Application and Reviews). 1998. Vol. 28. № 3. P. 392–403.
17. Jahanshah M., Meybodi M.R., Dehghan M. A New Approach for Task Scheduling in Distributed Systems Using Learning Automata // Proceedings of the IEEE International Conference on Automation and Logistics Shenyang. 2009. P. 62–67.

18. *Kasahara H., Narita S.* Practical multiprocessing scheduling algorithms for efficient parallel processing // *IEEE Trans. Computer.* 1984. Vol. C-33. № 11. P. 1023–1029.
19. *Kasahara H., Narita S.* Parallel processing of robot-arm control computation on a multimicroprocessor system // *IEEE J. Robotic J Automarion.* 1985. Vol. RA-1. № 2. P. 104–113.
20. *Kettimuthu R., Subrasani V., Srinivasan S., Gopalasamy T.* Selective Preemption Strategies for Parallel Job Scheduling // *Proc. Int. Conf. on Par. Proc. (ICPP).* 2002. P. 12–23.
21. *Laarhoven V., Peter J., Emile A., Lenstra J.* Job shop scheduling by simulated annealing // *Operations Research.* 1992. Vol. 40. № 1. P. 113–125.
22. *Merkle D., Middendorf M., Schmeck H.* Ant colony optimization for resource-constrained project scheduling // *IEEE transactions on evolutionary computation.* 2002. Vol. 6. № 4. P. 333–346.
23. *Narendra K.S., Thathachar M.L.* Learning automata: an introduction // Englewood Cliffs, NJ: Prentice Hall, 1989.
24. *O'Sullivan D., Unwin J.* Geographic Information Analysis, Second Edition. 2010. P. 432.
25. *Ramamoorthy C.V.* Optimal scheduling strategies in a multiprocessor system // *IEEE Trans. Computers.* 1972. P. 137–146.
26. *Shmueli E., Feitelson D.* Backfilling with Lookahead to Optimize the Packing of Parallel Jobs // *J. of Parallel and Distributed Computing.* 2005. Vol. 65. P. 1090–1107.
27. *Topcuoglu H., Hariri S., Wu M.* Performanceeffective and low-complexity task scheduling for heterogeneous computing // *IEEE transactions on parallel and distributed systems.* 2002. Vol. 13. № 3. P. 260–274.
28. *Thathachar M.* Stochastic automata and learning systems // Department of Electrical Engineering, Indian Institute of Science. 1990. P. 263–283.
29. *Tsafir D., Etsion Y., Feitelson D.* Backfilling using system-generated predictions rather than user runtime estimates // *IEEE Trans. Parallel and Distributed Systems.* 2007. Vol. 18. № 6. P. 789–803.
30. *Venter G., Sobieszczanski-Sobieski J.* Particle swarm optimization // *AIAA Journal.* 2003. Vol. 41. № 8. P. 1583–1589.
31. Заседание президиума Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам от 17.09.2018 г. URL: <http://government.ru/news/34001> (дата обращения: 13.05.2020).
32. Официальный сайт Национального проекта «Наука». URL: <https://futurerussia.gov.ru/all> (дата обращения: 10.05.2020).
33. Официальный сайт Национального проекта «Цифровая экономика». URL: <https://digital.gov.ru/> (дата обращения: 13.05.2020).
34. Рейтинг топ 50 суперкомпьютеров. URL: <http://top50.supercomputers.ru/> (дата обращения: 12.12.2020).

### Bibliography

1. *Mazalov V.V., Nikitina N.N.* Ocenka karakteristik algoritma Backfill pri upravlenii potokami zadach na vychislitel'nom klastere // *Vychislitel'nye tehnologii.* 2012. T. 17. № 5. P. 71–79.
2. *Mamojlenko S.N., Efimov A.V.* Algoritmy planirovaniya resheniya masshtabiruemykh zadach na raspredelennykh vychislitel'nykh sistemah // *Vestnik GOU VPO «SibGUTI».* 2010. № 2. P. 66–78.
3. *Volovich K.I.* Metody i algoritmy organizacii vychislitel'nogo processa v gibridnom vysokoproizvoditel'nom komplekse na osnove virtual'noj sredy ispolnenija: dis. ... kand. tehn. nauk: 05.13.15. M., 2019. 114 p.
4. *Tihomirov A.I.* Metody i sredstva organizacii sistemy upravlenija vychislitel'nymi zadaniyami v territorial'no raspredelennoy seti superkomp'yuternykh centrov kollektivnogo pol'zovaniya: dis. ... kand. tehn. nauk: 05.13.11. M., 2019. 143 p.

5. *Tjuljaeva E.O., Odincov I.O., Moskovskij A.A., Marmuzov G.V.* Tendencii razvitiya vychislitel'nyh uzlov sovremennyh superkomp'yuteroi // Vestnik JuUrGU. Serija «Vychislitel'naja matematika i informatika». 2019. P.92–114.
6. *Carrington L.A., Snavely A., Wolter N.* Performance Prediction Framework for Scientific Applications // Future Generation Computer Systems. 2006. Vol. 22. P. 336–346.
7. *Cirne W., Berman F.* A Comprehensive Model of the Supercomputer Workload // Proc. IEEE Fourth Ann. Workshop Workloads Characterization. 2001. P. 140–148.
8. *Chen C.L., Lee C.S., Hou E.S.* Efficient scheduling algorithms for robot inverse dynamics computation on a multiprocessor system // IEEE Trans. System Man Cybernetics. 1988. Vol. 18. P. 729–743.
9. *Clement M.J., Quinn M.J.* Analytical Performance Prediction on Multicomputers // J. Supercomputing. 1993. P. 886–894.
10. *Conway R.W., Maxwell W., Miller L.* Theory of Scheduling // Massachusetts Addison Wesley Publications. 1967. P. 304.
11. *Dinda P.A., O'Hallaron D.R.* Host Load Prediction Using Linear Models // Cluster Computing. 2000. Vol. 3. P. 265–280.
12. *Feitelson D., Rudolph L.H., Schwiegelshohn U., Wong P.* Theory and Practice in Parallel Job Scheduling // Theoretical Computer Science. 2004. P. 17–47.
13. *Gandhi T., Nitin, Alam T.* Quantum Genetic Algorithm with Rotation Angle Refinement for Dependent Task Scheduling on Distributed Systems // Tenth International Conference on Contemporary Computing (IC3). 2017. P. 12–15.
14. *Gonzalez M.J.* Deterministic processor scheduling // Computing Surveys. 1977. Vol. 9. № 3. P. 173–204.
15. *Hellstrom B., Kanal L.* Asymmetric mean-field neural networks for multiprocessor scheduling // Neural Networks. 1992. Vol. 5. P. 671–686.
16. *Hisao I., Murata T.* A multi-objective genetic local search algorithm and its application to flowshop scheduling // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Application and Reviews). 1998. Vol. 28. № 3. P. 392–403.
17. *Jahanshah M., Meybodi M.R., Dehghan M.* A New Approach for Task Scheduling in Distributed Systems Using Learning Automata // Proceedings of the IEEE International Conference on Automation and Logistics Shenyang. 2009. P. 62–67.
18. *Kasahara H., Narita S.* Practical multiprocessing scheduling algorithms for efficient parallel processing // IEEE Trans. Computer. 1984. Vol. C-33. № 11. P. 1023–1029.
19. *Kasahara H., Narita S.* Parallel processing of robot-arm control computation on a multimicroprocessor system // IEEE J. Robotic J Automarion. 1985. Vol. RA-1. № 2. P. 104–113.
20. *Kettimuthu R., Subrasani V., Srinivasan S., Gopalasamy T.* Selective Preemption Strategies for Parallel Job Scheduling // Proc. Int. Conf. on Par. Proc. (ICPP). 2002. P. 12–23.
21. *Laarhoven V., Peter J., Emile A., Lenstra J.* Job shop scheduling by simulated annealing // Operations Research. 1992. Vol. 40. № 1. P. 113–125.
22. *Merkle D., Middendorf M., Schmeck H.* Ant colony optimization for resource-constrained project scheduling // IEEE transactions on evolutionary computation. 2002. Vol. 6. № 4. P. 333–346.
23. *Narendra K.S., Thathachar M.L.* Learning automata: an introduction // Englewood Cliffs, NJ: Prentice Hall, 1989.
24. *O'Sullivan D., Unwin J.* Geographic Information Analysis, Second Edition. 2010. P. 432.
25. *Ramamoorthy C.V.* Optimal scheduling strategies in a multiprocessor system // IEEE Trans. Computers. 1972. P. 137–146.
26. *Shmueli E., Feitelson D.* Backfilling with Lookahead to Optimize the Packing of Parallel Jobs // J. of Parallel and Distributed Computing. 2005. Vol. 65. P. 1090–1107.
27. *Topcuoglu H., Hariri S., Wu M.* Performanceeffective and low-complexity task scheduling for heterogeneous computing // IEEE transactions on parallel and distributed systems. 2002. Vol. 13. № 3. P. 260–274.

28. *Thathachar M.* Stochastic automata and learning systems // Department of Electrical Engineering, Indian Institute of Science. 1990. P. 263–283.
29. *Tsafir D., Etsion Y., Feitelson D.* Backfilling using system-generated predictions rather than user runtime estimates // IEEE Trans. Parallel and Distributed Systems. 2007. Vol. 18. № 6. P. 789–803.
30. *Venter G., Sobieszczanski-Sobieski J.* Particle swarm optimization // AIAA Journal. 2003. Vol. 41. № 8. P. 1583–1589.
31. Zasedanie prezidiuma Soveta pri Prezidente Rossijskoj Federacii po strategicheskomu razvitiju i nacional'nym proektam ot 17.09.2018 g. URL: <http://government.ru/news/34001> (data obrashhenija: 13.05.2020).
32. Oficial'nyj sajt Nacional'nogo proekta «Nauka». URL: <https://futuresussia.gov.ru/all> (data obrashhenija: 10.05.2020).
33. Oficial'nyj sajt Nacional'nogo proekta «Cifrovaja jekonomika». URL: <https://digital.gov.ru/> (data obrashhenija: 13.05.2020).
34. Rejting top 50 superkomp'juterov. URL: <http://top50.supercomputers.ru/> (data obrashhenija: 12.12.2020).