



2. Полежаев П.Н., Ушаков Ю.А., Шухман А.Е. Система управления ресурсами для высокопроизводительных вычислений, основанная на использовании программно-конфигурируемой сети // "Системы управления и информационные технологии: научно-технический журнал", 2013. - №4(54). - С. 65-69.

Ю.С. Артамонов

ОСНОВНЫЕ ПОДХОДЫ ПРОГНОЗИРОВАНИЯ ДОСТУПНЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ В КЛАСТЕРНЫХ СИСТЕМАХ

(Самарский государственный аэрокосмический университет)

Введение. В настоящее время используется множество платформ для научных вычислений. В одном проекте может быть задействовано несколько различных окружений. Нехватка вычислительных ресурсов приводит к необходимости использования пакетных систем или же простых очередей заданий. Если в наличии есть разные по производительности и загруженности вычислительные ресурсы, то требуется оптимально выбрать окружение для расчётов. Критерием выступает отметка времени окончания вычислений.

При использовании суперкомпьютерной техники с пакетными системами пользователю требуется знать, когда его задание будет запущено. Если вычислительная задача выполняется в интерактивном режиме и выдаёт данные по мере вычислений, то ставится задача об оптимизации времени старта и завершения при условии, что вычислительных ресурсов достаточно для работы программы. Это может быть необходимо для последующего анализа выходных данных внешними по отношению к окружению инструментами.

Для решения этих задач требуются данные о загруженности вычислительных ресурсов и о профиле их использования. При прогнозировании следует принять во внимание как большой массив исторических данных по исполнению задач, так и тренд загрузки ресурсов. Значимыми могут являться и исторические данные, и тренд, также возможна ситуация когда влияние одного из факторов может быть ничтожно.

Выделим 3 задачи прогнозирования доступных вычислительных ресурсов:

1. Прогноз момента запуска вычислительной задачи.
2. Прогноз завершения вычислений задачи.
3. Прогноз доступных вычислительных ресурсов окружения.

Отдельно рассмотрим ограничения на общий объём вычислительных ресурсов.

1. Фиксированный объём вычислительных ресурсов.
2. Динамически изменяемый объём вычислительных ресурсов.

Случай с фиксированным объёмом вычислительных мощностей характерен для суперкомпьютерной техники, а примером случая с динамически изменяемыми вычислительными мощностями могут являться Desktop Grid системы.



Отдельно стоит сказать, что отказы оборудования не стоит рассматривать как ситуацию с динамически изменяемым объёмом ресурсов, поскольку известен верхний предел доступных мощностей. Случай с динамически изменяемым объёмом доступных ресурсов нельзя привести к случаю с фиксированными, поскольку изменения предела доступных мощностей носят случайный характер с неизвестным законом распределения.

Модель для прогнозирования. В работе будем рассматривать только случай с фиксированным объёмом вычислительных ресурсов, это позволит оценить работу методов прогнозирования на меньшем количестве входных данных, и данный случай более специфичен для работы с пакетными системами и суперкомпьютерной техникой. При этом отмена задач не допускается. Введём обозначения.

Ресурс - некоторый физический или логический ресурс системы, требуемый для запуска вычислительной задачи. Имеет набор численных параметров.

Параметр - некоторый значимый для вычислительных задач показатель ресурса.

Окружение - вычислительная система, включающая в себя ресурсы.

Ограничение - минимальные значения параметров ресурса, требуемые для запуска задачи.

Требование - набор ограничений, которым должны удовлетворять ресурсы окружения.

Следует отметить, что детали реализации пакетной системы, системы управления кластером или другого ПО не влияют на прогнозирование доступных ресурсов. Это позволяет строить алгоритм прогноза на данных универсальным образом, чтобы он был применим как для кластерных систем, так и для Desktop Grid.

Основные подходы прогнозирования доступных вычислительных ресурсов. К подходам прогнозирования доступных вычислительных ресурсов относятся:

- статистические модели;
- нейронные сети;
- распознавание периодических событий.

Для прогнозирования доступных вычислительных ресурсов подходят методы прогнозирования временных рядов и методы детектирования периодических событий (автокорреляционные функции, ряды Фурье, тригонометрическая интерполяция методом наименьших квадратов [1]). При этом проблемы возникают с шумом аperiodической природы – простои, отказы, снижение активности (например, в праздничные дни). Такие случаи плохо поддаются автоматическому анализу и требуются алгоритмы, которые могут игнорировать выбросы. Помимо шума сложность представляет поиск отрезков периодов с размытыми границами. Например, недельная активность на суперкомпьютере затухает по-разному в зависимости от набора задач в очереди.



Основные подходы прогнозирования времени запуска задач. Основными подходами прогнозирования времени запуска задач являются:

- вероятностные модели (QBETS [2]);
- параметрические модели (логнормальное распределение [3]);
- статистические модели (Статистический метод Дауни [4]);
- распознавание быстрых запусков (PQStar [5]);
- сравнение с задачами в истории [6];
- обучение по образцу (IBL [7]);
- прогнозирование по политике планировщика [8].

Все представленные подходы имеют свои достоинства и недостатки. Вероятностные модели позволяют получить прогноз с некоторой степенью уверенности, учитывая эффекты конкретного планировщика, но они дают при этом крайне консервативные результаты, завышая возможное время ожидания задачи в очереди.

Параметрические модели показали себя хуже QBETS, но в некоторых случаях они показывают сравнимый прогноз [5].

Статистический метод Дауни моделирует распределение времени выполнения при помощи логарифмической функции, при этом плохо учитываются выбросы (отказы) и периодические эффекты (простои для обслуживания).

Распознавание быстрых запусков может быть применено, если наблюдается тенденция к разделению времени ожидания задач в очереди на кластеры: до 1 часа, от 1 часа до 10 часов, более 10 часов. В таком случае можно извлечь пользу из информации о том, что задача будет ожидать в очереди менее часа. Недостаток распознавания быстрых запусков в малой информации, которую оно предоставляет.

Сравнение с задачами в истории не является самостоятельным методом, но часто используется в составе других (QBETS, PQStar), проблемы этого метода характерны для всех остальных. Основной недостаток в том, что метод не учитывает (или учитывает грубо) изменение конфигурации кластера и политики планирования. Например, в течение месяца была недоступна часть узлов кластера. В истории зафиксированы задачи, запускавшиеся в это время, их время ожидания будет учитываться при прогнозировании времени ожидания новых задач, хотя конфигурация кластера была изменена. Такие методы отдельно рассматривают задачу определения точек – границ актуальности истории.

Метод обучения по образцу также ищет подобные задачи в истории, при этом учитывает для каждой задачи помимо её параметров параметры всего окружения. В таком случае не требуется выявления границ актуальности истории, но возникает противоположная проблема – медленное накопление релевантной истории.

Прогнозирование по политике планировщика учитывает алгоритмы планирования и настройки планировщика. Метод не является универсальным и его сложно реализовывать, поскольку реализация может требовать существенных изменений даже при простом изменении параметров планировщика.



Результаты мониторинга задач на суперкомпьютере «Сергей Королёв». В рамках веб-сервиса Templet [9] мы собрали статистику исполнения задач на суперкомпьютере «Сергей Королёв» с ноября 2013 года по май 2014 года. Проанализировав профиль загрузки кластера (отрезок для узлов группы узлов `tmp`, `qdr` представлен на рис.1), мы можем сделать вывод о периодической природе загрузки. Как видно из профиля, загрузка снижается в выходные и праздничные дни и нарастает вновь в рабочие. Отрицательная ось графика показывает объём неработоспособных (отключенных) ресурсов. На оси абсцисс выделены жирными отрезками выходные дни.

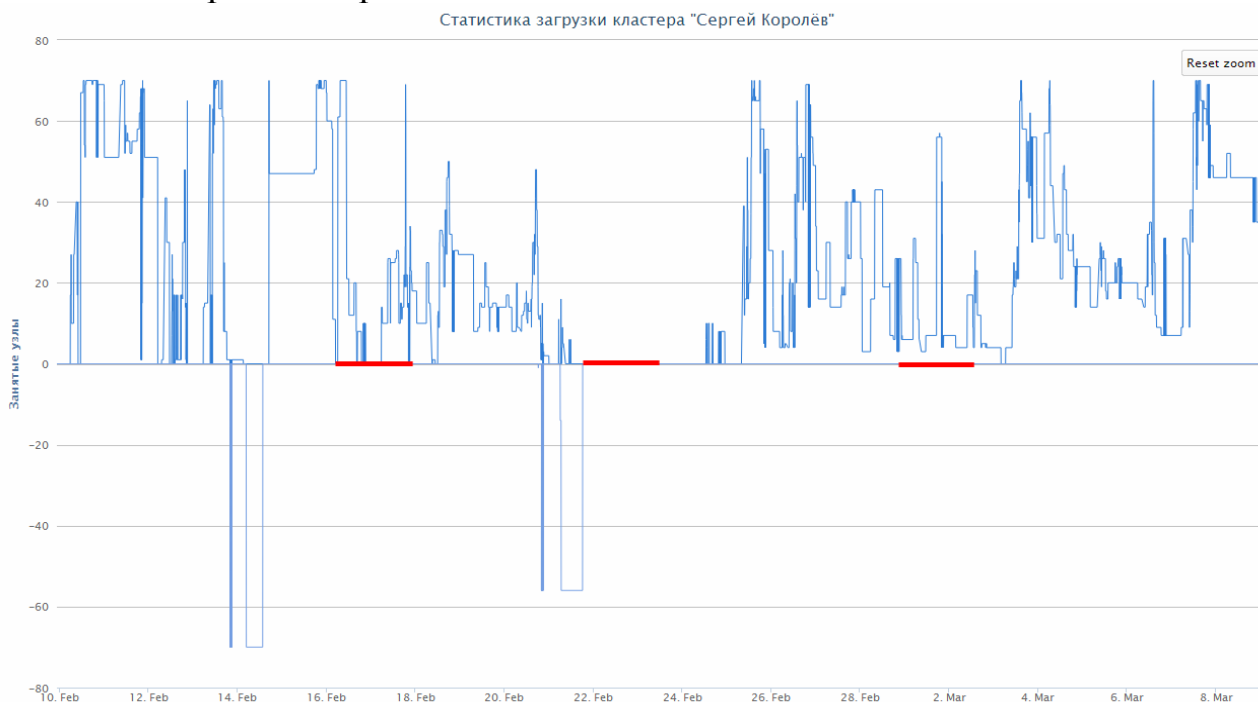


Рис. 1. Профиль загрузки суперкомпьютера «Сергей Королёв»

Помимо анализа загрузки мы построили график синтетической характеристики «суммарное время задач в очереди в секундах» (На рис. 2 представлен график для очередей пакетных задач `batch` и ПО моделирования `ansys`). Пунктирная черта показывает суммарное время задач в очереди равное 1 часу в расчёте на 164 узла суперкомпьютера. Из этого мы можем сделать вывод о том, что для прогнозирования времени запуска задач на суперкомпьютере «Сергей Королёв» может подойти метод распознавания быстрых запусков.

Применение результатов прогнозирования. Прогнозирование доступных ресурсов и момента запуска отдельных задач может быть применено для решения задач планирования:

1. выбора окружения для запуска задачи пользователя: с точки зрения длительности исполнения или с точки зрения надёжности систем;
2. построения плана размещения компонентов распределённого приложения;
3. оптимизации параметров запуска и объёмов входных данных для минимизации общего времени обработки.

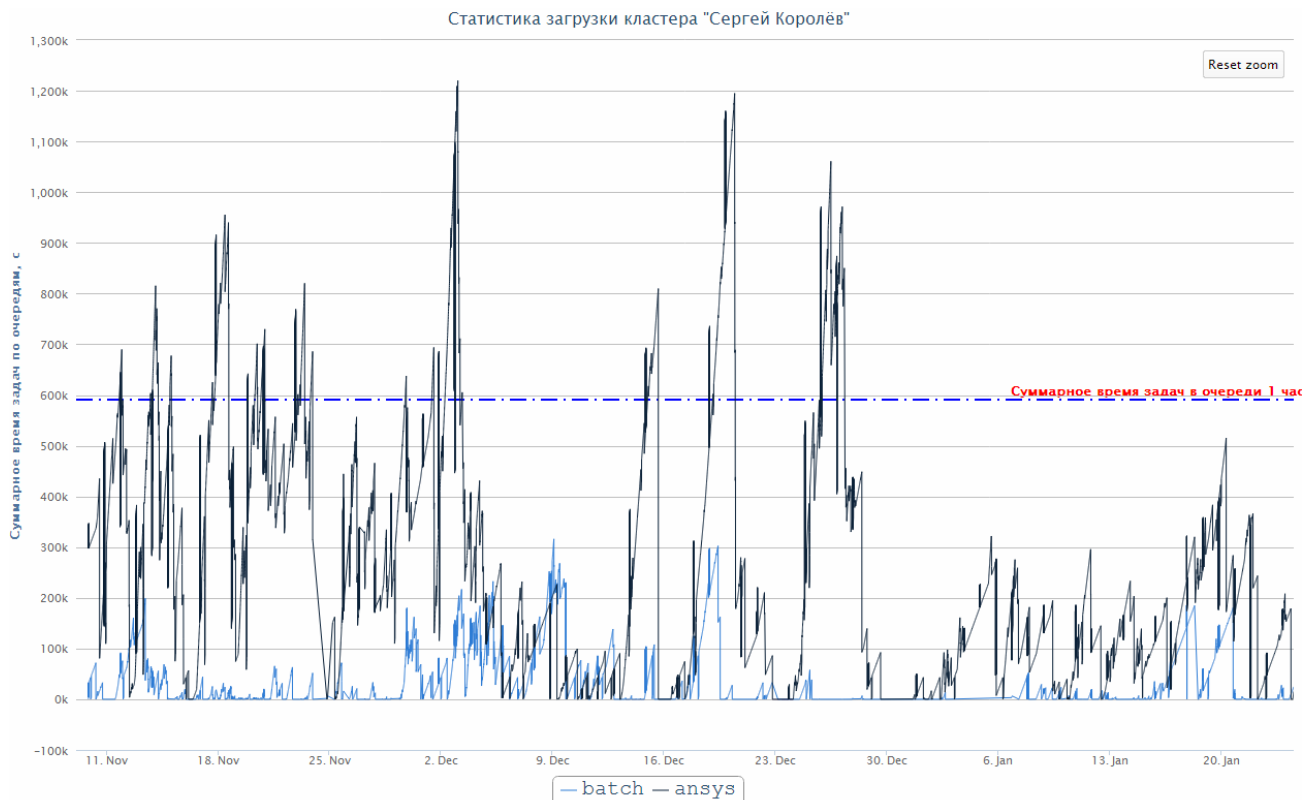


Рис. 2. Суммарное время задач в очереди

Прогнозирование доступных вычислительных ресурсов позволит выбирать целевое окружение для задачи с целью быстрого получения результатов, а также покажет, как долго конкретная задача может выполняться в целевом окружении. Практический результат также может быть получен и в области Desktop Grid систем, он позволит оценить надёжность определённой сети для расчётов с определённым фиксированным сроком исполнения.

Литература

1. Токмакова А.А. Выделение периодической компоненты из временного ряда [Текст] / А.А. Токмакова // Машинное обучение и анализ данных. — 2011. — № 1. — С. 40 - 50.
2. Nurmi D. QBETS: Queue Bounds Estimation from Time Series [Текст] / D. Nurmi, J. Brevik, R. Wolski // JSSPP 2007. LNCS, vol. 4942 Springer, Heidelberg. — 2008. — С. 76 - 101.
3. Brevik J. Predicting Bounds on Queuing Delay for Batch-Scheduled Parallel Machines [Текст] / J. Brevik, D. Nurmi, R. Wolski // PPOPP 2006: Proceedings of the Eleventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. — 2006. — С. 110 - 118.
4. Downey A.B Predicting Queue Times on Space-Sharing Parallel Computers [Текст] / A.B. Downey // IPPS 1997 Proceedings of the 11th International Symposium on Parallel Processing. — 1997. — С. 209 - 218.
5. Kumar R. Identifying Quick Starters: Towards an Integrated Framework for Efficient Predictions of Queue Waiting Times of Batch Parallel Jobs [Текст] / R.



Kumar, S. Vadhiyar // Job Scheduling Strategies for Parallel Processing, 16th International Workshop, JSSPP 2012 Shanghai. – 2012. – С. 196 - 215.

6. Smith W. Using Run-Time Predictions to Estimate Queue Wait Times and Improve Scheduler Performance [Текст] / W. Smith, V.E. Taylor, I.T. Foster // IPPS/SPDP 1999/JSSPP 1999: Proceedings of the Job Scheduling Strategies for Parallel Processing. – 1999. – С. 202 - 219.

7. Li H. Efficient Response Time Predictions by Exploiting Application and Resource State Similarities [Текст] / H. Li, D.L. Groep, L. Wolters // GRID 2005 Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing. – 2005. – С. 234 - 241.

8. Li H. Predicting Job Start Times on Clusters [Текст] / H. Li, D. Groep, J. Templon, L. Wolters // CCGRID 2004: Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid. – 2004. С. 301 - 308.

9. Артамонов Ю.С. Программный комплекс анализа многомерных динамических систем и процессов на суперкомпьютере «Сергей Королёв» [Текст] / Ю.С. Артамонов, Ю.П. Назаров, С.В. Востокин, А.В. Дорошин // Управление движением и навигация летательных аппаратов: сборник трудов XVI Всероссийского семинара по управлению движением и навигацией летательных аппаратов. - Самара: Издательство СНЦ РАН, 2013. - С. 60 - 63.

С.В. Болтанов

ПРОЕКТИРОВАНИЕ И ИССЛЕДОВАНИЕ ТИПОВОГО РЕШЕНИЯ «МЕТОД ПЕРЕМЕННЫХ НАПРАВЛЕНИЙ»: АЛГОРИТМЫ БАРЬЕРНОЙ СИНХРОНИЗАЦИИ

(Самарский государственный аэрокосмический университет)

В статье продолжают исследования типового решения «метод переменных направлений» и его эффективность, рассмотренные в [1] и [2].

При использовании типового решения «метод переменных направлений» возникла необходимость реализации и исследовании эффективности барьерной синхронизации и асинхронного ввода-вывода информации.

Синхронизация — это механизм, позволяющий наложить ограничения на порядок выполнения потоков. Путем синхронизации регулируется относительный порядок выполнения потоков и решается любой конфликт между потоками, который мог бы привести к нежелательному поведению программы. Синхронизация обеспечивает координацию выполнения потоков и управление совместно используемыми данными.

Основным свойством большинства параллельных итерационных алгоритмов является зависимость результатов каждой итерации от результатов предыдущей.

Причина, по которой в работе выбран данный способ синхронизации — барьерная синхронизация — в том, что, например, обычный мьютекс — это меха-