

Рис. 5. Генератор файлов

Литература

1. Куделькин В.А., Янников И.М. Структурная схема интеллектуальной интегрированной системы безопасности потенциально опасных объектов // Известия Самарского научного центра Российской академии наук. Том 17, №6(2), – 2015. – С. 726 – 728.
2. Куделькин В.А., Янников И.М. Алгоритм функционирования интеллектуальной интегрированной системы безопасности потенциально опасных объектов // Интеллектуальные системы в производстве № 3 (27) - Ижевск: Изд-во ИжГТУ, 2015. – С. 73 - 76.
3. Янников И.М., Куделькин В.А., Соболева Н.В. Функциональная модель интеллектуальной интегрированной системы безопасности потенциально опасных объектов // Интеллектуальные системы в производстве № 3 (27) - Ижевск: Изд-во ИжГТУ, 2015. – С. 77 - 82.
4. Организация охраны предприятия и физической защиты его объектов <http://bezopasnik.org/article/14.htm> (Дата обращения: 18.01.2016)
5. Н. В. Давидюк, С. В. Белов Формирование множества характеристик технических средств обнаружения, влияющих на задачу их выбора // Вестник АГТУ. Сер.: Управление, вычислительная техника и информатика. 2009. № 2. – С.110-113.



ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ И ТЕЛЕКОММУНИКАЦИИ

Ю.С. Артамонов

ПРОГНОЗИРОВАНИЕ ДОСТУПНЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ КЛАСТЕРА ПРИ ПОМОЩИ МОДЕЛИ EMMSР

(Самарский национальный исследовательский университет
имени академика С.П. Королёва)

В области научных вычислений остро стоит вопрос эффективности использования вычислительных ресурсов, поскольку сами ресурсы находятся в дефиците, а исследователи конкурируют за доступ к наиболее производительным окружениям. Под окружением понимается программно-аппаратный комплекс: совокупность физических ЭВМ, каналов связи, периферийных устройств и ПО, необходимого для работы системы. Нередко разработчикам проекта доступно не одно окружение для запуска вычислительных задач, а несколько. В этой ситуации важно грамотно выбрать окружение, в котором вычисления будут завершены как можно раньше.

подавляющее большинство современных кластеров и суперкомпьютеров используют пакетные системы для запуска задач, а это значит, что каждая задача перед своим запуском проходит через очередь пакетной системы. Время выполнения вычислений в различных окружениях может быть сравнимым, если сами окружения имеют схожую производительность, но время, которое проводит задача в очереди, может отличаться очень сильно, поскольку оно зависит от загруженности окружения и количества запускаемых задач [1]. Кроме того, время ожидания задачи в очереди может быть непредсказуемо для некоторых применяемых политик планирования задач.

Для прогнозирования объёма доступных ресурсов требуются данные о загруженности вычислительных ресурсов и о профиле использования. Под профилем использования понимается набор особенностей окружения, он может быть представлен историческими данными загрузки ресурсов [2]. При прогнозировании следует принять во внимание как большой массив исторических данных по исполнению задач, так и тренд загрузки ресурсов.

Модель прогнозирования EMMSР

Пусть задан временной ряд $Z(t) = Z(1), Z(2), \dots, Z(T)$. Набор последовательных значений $Z_t^M = Z(t), Z(t+1), \dots, Z(t+M-1)$, лежащий внутри исходного временного ряда, назовём выборкой длины M с моментом начала отсчёта t ;



$M \in \{1, 2, \dots, T\}$, $t \in \{1, 2, \dots, T - M + 1\}$. Две выборки одинаковой длины, принадлежащие одному временному ряду, свяжем через параметр временной задержки k : $Z_t^M = Z(t), \dots, Z(t + M - 1)$ и $Z_{t-k}^M = Z(t - k), \dots, Z(t - k + M - 1)$; $k \in \{1, 2, \dots, t - 1\}$. В работе [3] сформулировано ключевое свойство выборок, позволяющее прогнозировать значения временных рядов по известным выборкам: фактические выборки временного ряда могут иметь подобие с будущими выборками.

Подобием двух выборок назовём их свойство, заключающееся в том, что одна выборка может быть выражена через другую с помощью линейной зависимости $Z_t^M = \alpha_1 Z_{t-k}^M + \alpha_0 I^M + E^M$. Здесь α_1 и α_0 – коэффициенты, I^M – единичный вектор, E^M – вектор ошибок аппроксимации. Подобие выборок и его применение для прогнозирования временных рядов рассмотрено в работе [4], где автор приводит модель экстраполяции временных рядов по выборке максимального подобия (extrapolation model on most similar pattern, далее EMSSP).

Особенности модели EMSSP:

- модель относится к классу авторегрессионных моделей прогнозирования;
- модель работает со стационарными и нестационарными временными рядами;
- модель имеет один параметр M ;
- экстраполяция значений временного ряда производится за одну итерацию;

Для применения модели EMMSP должны выполняться условия:

- временной ряд равноотстоящий (значения процесса фиксируются через равные интервалы времени);
- временной ряд относится к классу рядов с длинной памятью;
- задача прогнозирования на P значений вперёд относится к классу краткосрочного или среднесрочного прогнозирования данного типа временного ряда;
- длина исходного временного ряда составляет не менее $500P - 700P$, где P – количество точек прогноза.

Применение модели EMMSP для прогнозирования загрузки ресурсов кластера «Сергей Королёв»

В рамках веб-сервиса Templet [5] мы собрали статистику исполнения задач на кластере «Сергей Королёв» с ноября 2013 г. по март 2016 г.

Мы опробовали модель EMMSP на данных по загрузке вычислительных ресурсов кластера. Модель работает недостаточно хорошо, чтобы применять её одну и даёт среднюю абсолютную ошибку ~10 узлов при максимальном числе узлов в группе – 70. Эту модель можно применять в комбинации с наивным прогнозом путём сдвига текущих значений ряда. Комбинацию моделей можно построить по эмпирическому правилу: если значение множителя подобия выборки α выходит за границы отрезка $[0, 3.0]$, то использовать сдвиг. На рисунке 1 представлен график загрузки узлов с фактическими (сплошная линия) и прогнозными (пунктирная линия) значениями для прогноза на 12 часов вперёд (12 значений, по одному в час). Пример участка, на котором лучший результат показывает модель прогнозирования по выборке наибольшего подобия, помечен



как EMMSP, а пример участка, на котором выполняется сдвиг, помечен как SHIFT.

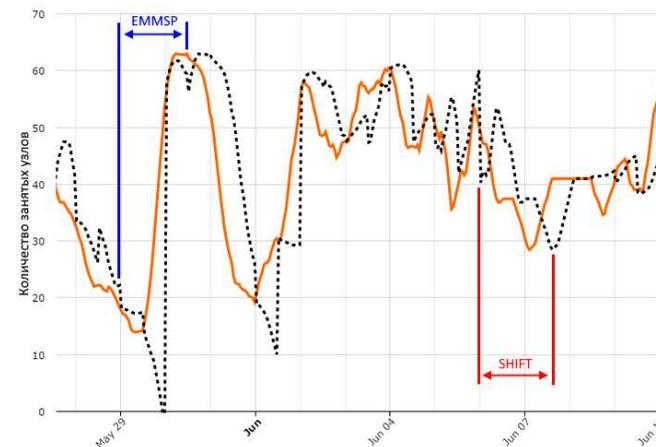


Рисунок 1 – Прогноз загрузки ресурсов кластера «Сергей Королёв»

По результатам тестирования модели мы можем сделать вывод о том, что модель хорошо себя показывает только на некоторых участках и её можно применять в комбинациях моделей путём адаптивной селекции или композиции [6].

Алгоритм прогнозирования будет интегрирован в сервис Templet Web, чтобы дать пользователям возможность оценить время запуска задачи.

Применение результатов прогнозирования

Результаты прогнозирования доступных ресурсов и момента запуска отдельных задач можно применять для решения задач планирования:

- выбор окружения для запуска задачи пользователя: обеспечить быстрый запуск, если задача интерактивная, или быстрое получение результатов;
- построение плана размещения компонентов распределённого приложения, если требуется построить распределённый конвейер вычислений;
- оптимизация параметров запуска и объёмов входных данных для минимизации общего времени обработки;
- снижение энергопотребления кластеров и центров обработки данных, когда ресурсы не используются;
- планирование увеличения парка серверов для облачных окружений, если загрузка ресурсов облака постоянно растёт.

С развитием облачных технологий вопросы энергоэффективности ресурсов выходят на первый план. Прогноз загрузки вычислительных ресурсов облака позволил бы отключать некоторое количество неиспользуемых ресурсов и сообщил бы о необходимости пополнения парка машин в случае роста загрузки.



Литература

1. 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.
2. 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.
3. Fernandez-Rodriguez, F. Nearest-Neighbour Predictions in Foreign Exchange Markets / F. Fernandez-Rodriguez, S. Sosvilla-Rivero, J. Andrada-Felix // Fundacion de Estudios de Economia Aplicada [Электронный ресурс]. – 2002, No.5. 36 p. URL: <http://documentos.fedea.net/pubs/dt/2002/dt-2002-05.pdf> (дата обращения 21.02.2016)
4. Чучуева, И.А. Модель прогнозирования временных рядов по выборке максимального подобия [Текст] / И.А. Чучуева // Дисс. на соиск. уч. ст. к. т. н., Москва. – 2012. – 155 с.
5. Артамонов, Ю.С. Применение облачного сервиса Templet Web при проведении лабораторных практикумов на суперкомпьютере «Сергей Королев» [Текст] / Ю.С. Артамонов, С.В. Востокин // X Международная научно-практическая конференция «Современные информационные технологии и ИТ-образование», МГУ, Москва, 2015. Том 2. – С. 409 - 414.
6. Лукашин, Ю.П. Адаптивные методы краткосрочного прогнозирования временных рядов [Текст] / Ю.П. Лукашин – Москва: Финансы и статистика, 2003 – 415 с.

И.П. Болодурина, Д.И. Парфёнов

СТРУКТУРНАЯ МОДЕЛЬ ПРОГРАММНО-УПРАВЛЯЕМОЙ ИНФРАСТРУКТУРЫ ВИРТУАЛЬНОГО ЦОД

(Оренбургский государственный университет)

В настоящее время, проблема эффективного использования вычислительных ресурсов в центрах обработки данных (ЦОД) является актуальной задачей. Современные коммуникационные технологии создали среду для многих критически важных бизнес-приложений и сервисов на основе облачных вычислений. В этой статье мы остановимся на вопросе моделирования инфраструктуры виртуальных центров обработки данных. В основе современных виртуальных ЦОД используется программно-конфигурируемая инфраструктура (SDI), на базе которой разворачиваются облачные приложения и сервисы по схемам IaaS и PaaS с использованием программно-конфигурируемой сети (SDN) [1-3]. Модель будет использована в дальнейшем для создания эффективных алгоритмов управления виртуальными ЦОД, включая алгоритмы планирования работы вирту-



альных машин, облачных приложений и сервисов, а также реконфигурации маршрутов сетевого трафика.

На рисунке 1 представлены уровневая схема программно-управляемой инфраструктуры виртуального ЦОД.



Рис. 1. Уровневая схема программно-управляемой инфраструктуры виртуального ЦОД

Основой структурной модели является программно-конфигурируемая сеть, которая может быть представлена в виде взвешенного ориентированного мультиграфа:

$$SDN = (Nodes, Links), \quad (1)$$

где $Nodes = \{Node_i\}_{i=1, N}$ – множество сетевых устройств (узлов / серверов / и т.д.); $Links = \{Link_j\}_{j=1, M}$ – множество дуг, представляющих собой сетевое соединение. Каждое сетевое устройство характеризуется следующим кортежем:

$$Node_i = (L_i, P, C, M, T) \quad (2)$$

где L_i – множество дуг, исходящих из вершины; $P: L_i \rightarrow Z^+ \cup \{0\}$ – функция, которая характеризует текущую задержку для каждой дуги; $C: L_i \rightarrow Z^+ \cup \{0\}$ – пропускная способность; $M: L_i \rightarrow Z^+ \cup \{0\}$ – максимальная пропускная способность; $T \in \{host, switch\}$ – тип устройства.

Структурная модель программно-конфигурируемые инфраструктуры ЦОД может быть определена как направленный мультиграф:

$$SDI = (Seg, Connect(t)), \quad (3)$$

где вершины графа $Seg = \{Seg_1, Seg_2, \dots, Seg_k\}$ – множество сетевых сегментов ЦОД; дуги графа $Connect(t) = \{(Seg_i, Seg_j)\}$ – направленная связь между сегментами.

Сегмент $Seg_k \in Seg$ программно-конфигурируемые инфраструктуры ЦОД представим в виде взвешенного неориентированного мультиграфа:

$$Seg_k = (Devices_k, Links_k, Flows_k(t)). \quad (4)$$

Вершины графа являются разбиением множества:

$$Devices_k = Nodes_k \cup Switches_k \quad (5)$$