

ПЕРСПЕКТИВНЫЙ ДЕЦЕНТРАЛИЗОВАННЫЙ ПОДХОД К БАЛАНСИРОВКЕ ЗАГРУЗКИ В МНОГОЯДЕРНЫХ ПРОЦЕССОРАХ

Н.О. Амелина, В.В. Васильев, О.Н. Граничин, В.И. Княев

В настоящее время при решении многих фундаментальных и прикладных задач используются многоядерные и многопроцессорные вычислительные устройства, реализующие кластерные и «облачные» технологии. При этом во многих случаях актуальной является задача балансировки загрузки узлов вычислительной сети, прямо влияющей на эффективность вычислений и оптимальную общую загрузку системы.

В работах А.Т.Вахитова, О.Н.Граничина и др. [1,2] исследована работоспособность алгоритмов стохастической аппроксимации с постоянным размером шага в условиях нестационарных функционалов качества (среднего риска). Показано, что такие алгоритмы можно применять для балансировки загрузки узлов централизованной вычислительной сети при доступности текущей зашумленной информации о длине очереди и производительности узлов, что в свою очередь можно использовать для разработки специальной программы – брокера загрузки.

Актуальность задачи балансировки подчёркивается интересом к этому вопросу в литературе – это, например, хорошо демонстрируют работы [3-5]. Авторы предлагают, например, использовать правило с равномерным случайным распределением и произвольное правило для балансировки загрузки сети [3]. Можно также рассматривать алгоритм балансировки загрузки гетерогенных систем с использованием мобильных агентов [4] или динамический алгоритм распределения заказов, который использует смешанную децентрализованную и централизованную политику [5].

Традиционно, задача балансировки нагрузки в многоядерной системе решается с помощью единого планировщика действующего на уровне ядра операционной системы. Однако, опираясь на опыт, пришедший из области «крупнозернистого» параллелизма (сильно связанные распределённые системы, GRID-системы) и мультиагентных технологий, можно организовать управление загрузкой процессора общего назначения с большим количеством ядер децентрализованными методами.

Для реализации этого принципа нами предлагается делегировать ответственность за балансировку нагрузки многоядерного процессора на каждое его ядро. Это возможно, если в дискретные моменты времени каждое ядро будет работать с собственной очередью загрузки и для каждого потока выполнения решать, продолжить ли его вычисление, или передать его соседнему ядру, уменьшив собственную очередь загрузки.

Н.О.Амелина и А.Л.Фрадков разработали способ, с помощью которого возможно переформулировать проблему балансировки как задачу о достижении консенсуса в мультиагентной сети [6]. В терминах данной задачи протокол, в соответствии с которым формируется решение о продолжении выполнения, называется «протоколом управления». В рамках такого случая наиболее эффективным из пока исследованных возможностей выступает так называемый «протокол локального голосования», для которого строгими математическими методами доказана асимптотическая сходимость к консенсусу (идеальному балансу).

Реализация подобного подхода включает либо использование отдельного аппаратного, либо специального программного планировщика на каждом ядре процессора. При этом если, например, в GRID-сетях балансировка нагрузкой осуществляется за счёт «перебрасывания» независимых атомарных заданий, то, переходя к организации параллельных вычислений на процессорах общего назначения, возникает сложность, связанная с трактовкой разбиения потока выполнения на отдельные задания. Конечно, наличие большого количества ядер позволяет в некотором смысле отойти от концепции вытесняющей многозадачности, так как в этом случае можно полностью отдавать отдельные ядра для выполнения на них привилегированных процессов (например, выделить отдельное ядро под работу кратковременного планировщика).

Заметим, что для «протокола локального голосования» доказана возможность достижения идеального баланса и, следовательно, получена оценка эффективности алгоритма балансировки загрузки узлов децентрализованной вычислительной сети при поступлении в каждый узел только зашумленной информации о длине очереди и производительности соседей. При использовании указанного алгоритма консенсус достигается, даже если топология связей является переменной (что избыточно для данной задачи), а информация от соседей доходит с задержкой по времени.

Рассмотрим модель системы распределения однотипных заданий между разными узлами (агентами) для параллельной работы с обратной связью. Пусть $N = \{1, \dots, n\}$ — набор интеллектуальных агентов (узлов), каждый из которых выполняет поступающие задания по принципу очереди. Задания поступают в систему в различные моменты времени и на разные узлы.

В каждый момент времени t состояние агента $i \in \{1, \dots, n\}$ описывается двумя характеристиками:

$q_{i,t}^i \geq 0$ – длина очереди из атомарных элементарных заданий узла i в момент времени t ;

$r_t^i > 0$ – производительность узла i в момент времени t .

Изменения состояний агентов определяются следующим уравнением:

$$q_{t+1}^i = q_t^i - r_t^i + z_t^i + u_t^i,$$

где $i = 1, \dots, n$; $t = 0, 1, 2, \dots$; z_t^i – новое задание, поступившее на узел i в момент времени t , u_t^i – управляющее воздействие в тот же момент времени.

Будем считать, что для формирования стратегии управления каждый узел i в момент времени t имеет следующую информацию:

- зашумленные данные о своей длине очереди $y^{i,i}_t$,
- зашумленные наблюдения о длинах очередей соседей $y^{i,j}_t$,
- данные о своей производительности r_t^i и о производительностях узлов соседей r_t^j .

Будем рассматривать две постановки задачи: стационарную и нестационарную.

Стационарный случай

Все задания поступают в систему на разные узлы в начальный момент времени и производительности узлов не меняются с течением времени.

Если все задания выполняются только тем агентом, которому они поступили, то время реализации всех заданий определяется как максимум из отношений загруженностей в начальный момент времени к соответствующим производительностям.

Нестационарный случай

Новые задания могут поступать в систему на любой из n узлов в различные моменты времени t и производительности узлов могут изменяться с течением времени. Для момента времени t определим T_t - время до окончания выполнения всех заданий на всех узлах.

Если в стационарном случае, начиная с момента времени t , задания не перераспределяются между узлами, то время реализации всех заданий определяется как максимум из отношений загруженностей в момент времени t к соответствующим производительностям.

Будем называть отношение загруженности в момент времени t к соответствующей производительности узла i состоянием узла i в момент времени t . Цель управления - минимизировать T_t . Для минимизации времени выполнения всех заданий естественно использовать протокол перераспределения заданий с течением времени. Это позволит увеличить пропускную способность системы и уменьшить время выполнения заказов в системе.

В работе [6] доказана следующая лемма об оптимальной стратегии управления: «В стационарном случае из всех возможных вариантов распределения общего количества заданий, необработанных к моменту времени t , наименьшее время работы системы соответствует тому, при котором состояния всех узлов одинаковые». Следствие из этой леммы: «Цель управления – достижение консенсуса в сети - соответствует оптимальному распределению заданий между узлами в стационарном случае».

Если рассматривать актуальную задачу о поддержке равномерной загрузки всех узлов сети, то для её решения можно предлагается использовать протокол «локального голосования». При его реализации на каждой итерации состояние каждого из узлов изменяется прямо пропорционально в зависимости от суммы разностей информации о его состоянии с информацией о состояниях его соседей, получаемых с задержками и помехами. Для этого случая установлены достаточные условия, обеспечивающие достижение консенсуса, т.е. оптимальную загрузку узлов вычислительной сети [6].

Отметим, однако, что применение протокола «локального голосования» на практике требует большей аккуратности. В рассмотренной ситуации идеализированно предполагается мгновенный обмен соответствующими потоками входных данных. Кроме того, надо дополнительно проверять согласованность пересылки пакетов, так как возможны различные коллизии за счет задержек и получения информации с помехами. В частности, использование протокола «локального голосования» на практике в задаче балансировки загрузки узлов децентрализованной вычислительной сети в действительности предполагает дополнительные согласования размеров пересылаемых пакетов между агентами.

При перераспределении ресурсов (либо заданий между узлами) не допускается ни «перерасход», ни «недорасход» ресурсов. Для выполнения этого условия в каждом узле происходят дополнительные проверки и согласования с соседями. По протоколу локального голосования каждый узел определяет, сколько он «отдаст» или «примет» заданий. Далее те узлы, которые готовы принять задания посылают запросы соседям о том, сколько они фактически готовы передать в данный момент. В ответ на эти запросы каждый узел «принимающий» задания дает подтверждение сколько он примет от того или иного узла заданий и согласовывает это количество с ним (при этом каждый узел ориентируется на свои текущие значения рассчитанных управляющих значений, которые рекомендуются протоколом локального голосования). Предполагается, что процедура согласования и передачи заданий по времени занимает существенно меньше времени, чем длительность одного такта работы динамической сети [7].

Авторы убеждены, что подобная децентрализованная система с одной стороны более естественна, с другой, позволяет избежать некоторых накладных расходов и проблем с надёжностью, возникающих при использовании единого брокера загрузки.

ЛИТЕРАТУРА:

1. А.Т. Вахитов А.Т., О.Н. Граничин О.Н., Гуревич Л.С. Алгоритм стохастической аппроксимации с пробным возмущением на входе в нестационарной задаче оптимизации // Автоматика и телемеханика. 2009. № 11. С. 70-79
2. А.Т. Вахитов, О.Н. Граничин, М.А. Панышенков. Методы оценивания скорости передачи данных в GRID // Нейрокомпьютеры: разработка, применение. 2009. № 11. С. 45-52
3. T.A. Friedrich, T.B. Sauerwal, D.C. Vilenchik. Smoothed analysis of balancing networks // Random Structures and Algorithms. Aug. 2011. Vol. 39. No.~1. P. 115-138
4. H. Li. Load balancing algorithm for heterogeneous P2P systems based on Mobile Agent // Proc. of ICEICE 2011. 2011. P. 1446-1449
5. M.T. Kechadi, I.K. Savvas. Dynamic task scheduling for irregular network topologies // Parallel Computing. 2005. Vol. 31. No. 7. P.757-776
6. Н.О. Амелина, А.Л. Фрадков. Достижение консенсуса в децентрализованной стохастической динамической сети при неполной информации и задержках в измерениях // Автоматика и Телемеханика. 2012
7. Н. Амелина, А. Лада, И. Майоров, П. Скобелев, А. Царев. Исследование моделей организации грузовых перевозок с применением мультиагентной системы адаптивного планирования грузовиков в реальном времени, Проблемы управления. № 6. С. 31-37, 2011