

границам верхних поверхностей цилиндрических объектов. Точки определяются автоматически в результате серии геометрических построений на изображении. Для выделения на изображении границ верхних поверхностей объектов используется оконтуривание детектором Кэнни с последующей аппроксимацией пикселей.

УДК 681.518

В.А. Литвиненко, С.А. Ховансков

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ В СЕТИ МЕТОДОМ КОЛЛЕКТИВНОГО ПРИНЯТИЯ РЕШЕНИЯ

В настоящее время множество задач требуют выполнения большого объема вычислений за минимальное время. Самым популярным решением этой проблемы в настоящее время является применение распределенной вычислительной системы. Это многомашинные системы, мультипроцессорные системы. Наиболее перспективным направлением является использование обычной компьютерной сети для организации многомашинной системы выполняющей распределенные вычисления.

При организации выполнения распределенных вычислений в обычной компьютерной сети решаются такие основные задачи как организация совместной работы компьютеров в сети, распределение между ними вычислительных модулей, поддержка работоспособности системы, подбор компьютеров в сети.

Методов организации такой многомашинной вычислительной системы множество. Основным недостатком этих методов является использование труда программиста на конфигурацию и настройку системы под конкретную задачу, а также расход вычислительных ресурсов и времени на оптимизацию системы, способной удовлетворять требованиям к отводимым временным ресурсам [1].

Основной задачей было создать метод, позволяющий свести к минимуму подготовительные этапы для решения любой задачи, использовать в качестве вычислительных центров любые компьютеры сети, оптимизировать время выполнения задачи за счет индивидуального подбора компьютера для решения конкретного вычислительного блока. В то же время создаваемая система должна быть работоспособной при любом наборе компьютеров, как по количеству, так и по производительности, обладать высокой живучестью - не терять работоспособность и отведенные под задачу временные ресурсы при динамическом изменении набора компьютеров, составляющих вычислительную систему.

Наиболее перспективным путем организации выполнения распределенных вычислений в компьютерной сети является использование мультиагентной системы, реализующей метод коллективного принятия решения.

Сама по себе мультиагентная система – это набор агентов, каждый из которых представляет программный модуль и помещается на отдельном компьютере. Агент выполняет управление только своим компьютером и поэтому его работа является независимой. Он организует решение задач на своем компьютере, иницирует обмен данными с компьютерами других агентов, выполняет обработку полученной от других агентов информации и на ее основе принимает решения.

Все агенты образуют одноранговый набор и работают по одному и тому же алгоритму. Отличительной особенностью мультиагентной системы является способность увеличивать число образующих ее компьютеров путем передачи копии программного модуля агента в свободные компьютеры. Благодаря тому, что аген-

ты работают по одному алгоритму, возможно решение поставленных задач методом принятия коллективного решения.

Для организации распределенной вычислительной системы на основе мультиагентной системы, в которой при организации выполнения любой задачи были решены перечисленные выше основные задачи, и был разработан алгоритм работы агента мультиагентной системы.

Структура графа решаемой задачи представляет собой совокупность подмножеств связанных вершин. Эти подмножества образуют иерархическую структуру с несколькими уровнями.

Самым верхним уровнем является вершина графа, получающая выходные данные. Подмножества образуют узлы нижнего уровня и узлы высокого уровня, которым передаются данные.

Другие подмножества содержат вершину верхнего уровня вместе с связанными с ней вершинами более нижнего уровня.

Самым нижним уровнем является вершина, принимающая входные данные.

Вся информация передается между агентами пакетами. Обработку одной порции данных можно считать одним шагом процесса обработки всей системой. Агенту, передавшему обработанные данные, поступает новая порция данных, и выполняется следующий шаг обработки. Передача данных в сети от входного компьютера к выходному выполняется в пульсирующем режиме.

Обмен данными между агентами включает в себя не только передачу обрабатываемых данных, но и служебную информацию.

После пересылки обработанных данных компьютеру агента более высокого уровня выполняется обмен между агентами приготовленной заранее служебной информацией. Обмен служебной информацией производится между агентами, расположенными только на компьютерах одного либо соседних подмножеств.

Служебная информация используется для минимизации времени обработки пакета данных на одном шаге. Время обработки данных в компьютере зависит от аппаратных параметров компьютера и свойств программного обеспечения, а также от характера обработки данных.

В системе временем начала обработки данных на компьютерах агентов текущего уровня будет считаться время приема последнего требуемого пакета данных от агентов нижнего уровня связанных с ними. Соответственно время, за которое на этом пути появятся обработанные на предыдущем уровне данные, и будет временем выполнения предыдущего шага.

Оценка времени выполнения вычислений на одном компьютере производится на основании работы реальной задачи, поэтому эта оценка является наиболее эффективной для выполнения оптимизации системы.

На графе задачи существует несколько маршрутов передачи обработанных данных от входного компьютера к выходному. Из них можно выделить тот путь передачи данных от агента самого нижнего уровня к агенту самого высокого уровня, который и определяет время обработки данных всей вычислительной системой. Такой путь называется критическим путем. $T_{кр} = \max(T_1, T_2, \dots, T_i, \dots, T_k)$,

где k - количество путей между входным агентом $a_{вх}$ и выходным агентом $a_{вых}$.

В предполагаемом алгоритме оптимизируется время выполнения системой задачи путем подбора назначения графа задачи на граф сети.

Все множество вершин графа задачи разбивается на непересекающиеся подграфы путем их раскраски G . Для раскраски графа используются функциональные признаки вычислительных блоков.

Сначала оптимизация выполняется одновременно внутри каждого подграфа между циклами обработки данных. Затем выполняется оптимизация между соседними подграфами.

При этом время, затрачиваемое агентами подграфа на оптимизацию, не зависит от мощности подграфа, что позволяет отвести под процесс оптимизации во всех подграфах фиксированный промежуток времени.

В следующий промежуток обмена информацией между агентами соседних подграфов выполняется аналогичная оптимизация в соседних подграфах.

Процесс оптимизации между агентами в одном подграфе многократно чередуется с оптимизацией между агентами в соседних подграфах. Этот процесс будет постоянно повторяться, поскольку процесс будет выполняться при каждой передаче данных между агентами вычислительных блоков во время выполнения задачи, то он будет вестись постоянно на фоне выполнения задачи. Время, затрачиваемое на оптимизацию, будет сведено к минимуму.

Передача данных производится от нижнего уровня на более верхние согласно графу задачи G . Самым верхним уровнем является выходной компьютер. Все множество P компьютеров с агентами, реализующие граф задачи, разбиты на подмножества вершин по уровням $P = \{P_1, P_2, \dots, P_n\}$, которые, в свою очередь, могут состоять из более мелких подмножеств $P_n = \{p^{n,1}, p^{n,2}, \dots, p^{n,b}\}$. Любое подмножество вершин $p^{n,b}$ состоит из агентов на компьютерах $(k_1^{n,b}, k_2^{n,b}, \dots, k_m^{n,b})$, соединенные по графу задачи с одним из компьютеров верхнего уровня $k_i^{n+1,c}$.

Подмножеству компьютеров $(k_1^{n,b}, k_2^{n,b}, k_3^{n,b}, \dots, k_m^{n,b})$ перед выполнением первого шага задачи назначено подмножество $(g_1^{n,b}, g_2^{n,b}, g_3^{n,b}, \dots, g_m^{n,b})$ графа задачи G .

После выполнения первого шага задачи на компьютерах подмножества $p^{n,b}$ осуществляется передача результатов вычислений вместе с управляющей информацией агенту на компьютере $k_i^{n+1,c}$. Собирается информация от каждого агента о времени выполнения этого шага $(t_1^{n,b} \in k_1^{n,b}, t_2^{n,b} \in k_2^{n,b}, t_3^{n,b} \in k_3^{n,b}, \dots, t_m^{n,b} \in k_m^{n,b})$ и выбирается компьютер $k_y^{n,b}$, имеющий $t_{\min 1}^{n,b} = \min(t_1^{n,b}, t_2^{n,b}, \dots, t_m^{n,b})$. После поступления управляющей информации от агента компьютера $k_i^{n+1,c}$ агенты подмножества $p^{n,b}$ получают информацию, принятую от всех компьютеров $p^{n,b}$. Выполняется второй шаг решения задачи.

После выполнении второго шага задачи на компьютерах подмножества $p^{n,b}$ осуществляется передача информации агенту на компьютере $k_i^{n+1,c}$, как и на первом шаге. Также собирается информация о времени выполнения этого шага всеми компьютерами и выбирается компьютер $k_x^{n,b}$, имеющий $t_{\min 2}^{n,b} = \min(t_1^{n,b}, t_2^{n,b}, \dots, t_m^{n,b})$.

Компьютер $k_i^{n+1,c}$ оценивает время завершения первого шага решения и второго. Если $t_{\min 1}^{n,b} > t_{\min 2}^{n,b}$, то принимается решение о выполнении нового переназначения агентам подмножества $p^{n,b}$, аналогичное выполненному перед вторым шагом.

Одновременно, точно по такому алгоритму, выполняются переназначения и во всех подмножествах всех уровней.

Вся информация собирается у агента $a_{\text{вылх}}$. Этой информации достаточно, чтобы определить критический путь и выделить в нем подмножество компьютеров, определяющее время выполнения шага системой. Агент $a_{\text{вылх}}$ выбирает подмножество, имеющее минимальное время выполнения шага и передает указание подобрать компьютеры для обмена между подмножествами так, чтобы максимальное критическое время системы сократилось.

Алгоритм агента

1. Прием информации от агентов уровня $n - 1$.
2. Прием служебной информации от агентов уровня $n + 1$.
3. Определение критического пути.
4. Подбор варианта обмена для агента на $T_{кр}$.
5. Все варианты обмена использованы? Да, перейти к п.7, нет п.6.
6. Передача служебной информации агентам уровня $n - 1$.
7. Обработка данных..
8. Обработка данных закончена? Да, перейти к п.9, нет п.7
9. Передача всей информации агентам уровня $i + 1$.
10. Переход к п.1.

Алгоритма назначения вершин графа задачи G компьютерам сети под управлением мультиагентной системой позволяет выполнять оптимизацию при любом первоначальном назначении графа G на сеть. Этот алгоритм не требует предварительной оптимизации времени решения компьютерами сети. Оптимизация выполняется уже во время решения самой задачи, используя для этого реальные данные. Позволяет обеспечить высокую степень живучести системы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *В.А. Литвиненко, С.А. Ховансков* Организация распределенных вычислений на основе мультиагентного подхода // Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». – Таганрог: Изд-во ТРТУ, 2007. №1(73). – С.246-250