

Исследование различных способов построения алгоритмов мета-планирования ресурсов центра обработки данных

Ледовский М. В.

Аннотация— Данная работа посвящена исследованию различных способов построения алгоритмов метапланирования ресурсов центров обработки данных. Дано описание модели запроса и центра обработки данных, в рамках модели сформулирована математическая постановка задачи распределения запросов по локальным планировщикам ресурсов ЦОД. Предложены способы построения алгоритмов распределения запросов.

Ключевые слова— Мета-планировщик ресурсов, облачные вычисления, центр обработки данных.

I. ВВЕДЕНИЕ

Рассмотрим группу, состоящую из нескольких центров обработки данных. Каждый из центров обработки данных обладает определенным числом вычислительных узлов, каждый из которых характеризуется производительностью, числом систем хранения данных, каждая из которых характеризуется объемом, а также числом каналов связи, каждый из которых характеризуется заданной пропускной способностью. На основе этих характеристик и информации о характеристиках выполняемых в ЦОД заявок для каждого типа ресурсов может быть вычислено значение суммарных остаточных ресурсов. Также каждый из ЦОДов обладает локальным планировщиком ресурсов. В задачу локального планировщика ресурсов входит непосредственное размещение запросов на физические ресурсы ЦОД для обработки.

В группу ЦОДов поступают запросы, которые необходимо разместить в ЦОДы для обработки. Эти запросы составляют пул запросов. Каждый запрос из пула характеризуется определенным количеством виртуальных машин, для каждой из которых определены значения требуемой производительности, количеством storage-элементов, каждый из которых требует заданный объем памяти, количеством виртуальных каналов связи, для каждого из которых определены значения пропускной способности. Суммируя эти характеристики для каждого типа ресурсов можно определить требуемы для пула суммарные характеристики.

Требуется распределить запросы из пула запросов между локальными планировщиками ресурсов ЦОД

таким образом, чтобы количество размещенных запросов было максимальным.

II. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

A. Модель запроса

Запрос – единица данных, поступающих на вход мета-планировщика ресурсов. Множество запросов образуют пул запросов.

Запрос i из пула задается графом [1]

$$HR_i = (W_i \cup S_i \cup E_i) \quad (1)$$

где

W_i - множество виртуальных машин, реализующих приложения;

S_i - множество storage-элементов;

E_i - множество виртуальных каналов передачи данных между виртуальными машинами и storage-элементами.

Для каждого типа ресурсов в запросе определены функции, задающие значения требуемой производительности.

На множестве W_i определена функция $vh(w)$, задающая производительность виртуальной машины (оп./с.); на множестве S_i определена функция $uh(s)$, задающая объем памяти storage-элемента (байт); на множестве E_i определена функция $th(e)$, задающая пропускную способность виртуального канала передачи данных (байт/с.);

B. Модель ЦОД

Модель ЦОД задается графом

$$HDC_j = (X_j \cup Y_j \cup Z_j) \quad (2)$$

где

X_j - множество вычислительных узлов ЦОД,

Y_j - множество хранилищ данных,

Z_j - множество физических каналов связи.

Для каждого типа ресурсов определены функции, задающие значения остаточных ресурсов ЦОД.

На множестве X_j определена функция $v(x)$, задающая производительность вычислительных узлов ЦОД (оп./с.); на множестве Y_j определена функция $u(y)$, задающая объем физической памяти (байт); на множестве Z_j определена функция $r(z)$, задающая пропускную способность каналов связи (байт/с.).

C. Размещение запроса на ресурсы ЦОД

Каждый центр обработки данных обладает локальным планировщиком ресурсов, которому передается пул запросов для размещения. С математической точки зрения размещение запросов на ресурсы ЦОД является

отображением [2]

$$A: HR_i \rightarrow HDC_j = \{W \rightarrow X, S \rightarrow Y, E \rightarrow Z\} \quad (3),$$

которое удовлетворяет следующим условиям:

- 1) виртуальная машина w может быть назначена на выполнение на вычислительном узле p , если

$$\sum_{w \in W_p} v(w) \leq vh(p)$$

выполняется условие , здесь W_p - множество виртуальных машин, назначенных на выполнение на вычислительном узле p . То есть производительность вычислительного узла должна быть не меньше требуемой суммарной производительности назначенных на него виртуальных машин и объем памяти вычислительного узла должен быть не меньше требуемой суммарной памяти назначенных на него виртуальных машин. Планирование выполнения виртуальных машин на вычислительном узле осуществляется локальным планировщиком вычислительного узла. Локальный планировщик должен гарантированно обеспечивать для каждой виртуальной машины требуемую производительность;

- 2) виртуальный канал e может быть отображен на физический канал l , если выполняется условие $\sum_{e \in E_k} r(e) \leq th(k)$, здесь E_k - множество

виртуальных каналов, отображенных на физический канал l ;

- 3) storage-элемент s может быть размещен в хранилище данных m , если выполняется

$$\sum_{s \in S_m} u(s) \leq uh(m)$$

условие , здесь S_m - множество storage, размещенных в хранилище данных m , а также тип хранилища данных совпадает с типом storage-элемента.

D. Модель остаточных ресурсов ЦОД

Модель остаточных ресурсов каждого ЦОД задается графом остаточных ресурсов H_{res} , для которого переопределены функции

$$vh_{res}(p) = vh(p) - \square_{w \in W_p} v(w),$$

$$uh_{res}(m) = uh(m) - \square_{s \in S_m} u(s), \quad (4)$$

$$th_{res}(k) = th(k) - \square_{e \in E_k} r(e)$$

E. Суммарные остаточные ресурсы ЦОД

На основе графа остаточных ресурсов ЦОД необходимо вычислить значения суммарных остаточных ресурсов ЦОД:

- 1) Суммарная остаточная производительность вычислительных узлов количеством :

$$\sum_{a2=1}^{A2_j} v(p_{j,a2}) \quad (5)$$

- 2) Суммарный объем физической памяти количеством :

$$\sum_{b2=1}^{B2_j} u(m_{j,b2}) \quad (6)$$

- 3) Суммарная остаточная пропускная способность каналов связи количеством :

$$\sum_{c2=1}^{C2_j} r(l_{j,c2}) \quad (7)$$

На основе параметров (5)-(7) составляется вектор суммарных остаточных ресурсов ЦОД:

Вектор суммарных остаточных характеристик j – го ЦОД:

$$\overline{HC}_j = \left(\sum_{a2=1}^{A2_j} v(p_{j,a2}), \sum_{b2=1}^{B2_j} u(m_{j,b2}), \sum_{c2=1}^{C2_j} r(l_{j,c2}) \right) \quad (8)$$

F. Суммарные требуемые ресурсы входного пула запросов

На основе графа требуемых ресурсов (1) необходимо вычислить значения суммарных требуемых ресурсов пула запросов:

- 1) Необходимая суммарная производительность всех виртуальных машин в поступившем пуле запросов ($A1$ – общее количество виртуальных машин в пуле запросов, каждый запрос из пула требует виртуальные машины в количестве $A1_i$):

$$\sum_{i=1}^n \sum_{a1=1}^{A1_i} vh(w_{i,a1}) \quad (9)$$

- 2) Необходимый суммарный объем памяти всех storage-элементов в поступившем пуле запросов ($B1$ – общее количество storage-элементов в пуле запросов, каждый запрос из пула требует storage-элементы в количестве $B1_i$):

$$\sum_{i=1}^n \sum_{b1=1}^{B1_i} uh(s_{i,b1}) \quad (10)$$

- 3) Необходимая суммарная пропускная способность всех каналов связи в поступившем пуле запросов ($C1$ – общее количество виртуальных каналов связи в пуле запросов, каждый запрос из пула требует виртуальные каналы связи в количестве $C1_i$):

$$\sum_{i=1}^n \sum_{c1=1}^{C1_i} th(e_{i,c1}) \quad (11)$$

Вектор суммарных требуемых ресурсов пула запросов:

$$\overline{HR} = \left(\sum_{i=1}^n \sum_{a1=1}^{A1_i} vh(w_{i,a1}), \sum_{i=1}^n \sum_{b1=1}^{B1_i} uh(s_{i,b1}), \sum_{i=1}^n \sum_{c1=1}^{C1_i} th(e_{i,c1}) \right) \quad (11)$$

Алгоритм распределения запросов по центрам обработки данных (мета-планировщик) получает в качестве входных данных вектор суммарных требуемых ресурсов пула запросов (11) и вектор суммарных остаточных ресурсов ЦОД (8).

Необходимо распределить запросы из пула запросов между локальными планировщиками ресурсов ЦОД таким образом, чтобы количество размещенных запросов было максимальным.

III. АЛГОРИТМ МЕТАПЛАНИРОВАНИЯ

При поступлении пула запросов вычисляются необходимые характеристики: суммарные требуемые ресурсы пула запросов и суммарные остаточные характеристики ЦОДов. Далее алгоритм определяет, возможно ли полностью разместить пул запросов в один из доступных ЦОДов. В случае, если подходящий ЦОД найден, то происходит передача поступившего пула запросов целиком планировщику ЦОД. В противном случае поступивший пул запросов делится на подпулы, затем подпул передается планировщику ЦОД для размещения. Если планировщик ЦОД не смог разместить запросы, то эти запросы попадают в общий входной пул.

A. Проверка на способность ЦОД полностью разместить поступивший пул запросов

Если среди ЦОДов найдется тот j -ый ЦОД, у которого суммарные остаточные характеристики **больше или равны** требуемым суммарным характеристикам поступившего пула запросов (13), то в этот ЦОД отправляется пул запросов на размещение **целиком**.

Для j -го ЦОДа, $j = \overline{1, k}$:

$$\begin{cases} \sum_{a2=1}^{A2j} v(p_{j,a2}) \geq \sum_{i=1}^n \sum_{a1=1}^{A1i} vh(w_{i,a1}) \\ \sum_{b2=1}^{B2j} u(m_{j,b2}) \geq \sum_{i=1}^n \sum_{b1=1}^{B1i} uh(s_{i,b1}) \\ \sum_{c2=1}^{C2j} r(l_{j,c2}) \geq \sum_{i=1}^n \sum_{c1=1}^{C1i} th(e_{i,c1}) \end{cases} \quad (13)$$

Если условие (13) не выполняется, то необходимо разделить пул запросов на подпулы.

В случае, если на размещения пула целиком претендуют больше одного ЦОДа, то для каждого из них вычисляется вектор разности (14):

$$\overline{\delta} = \begin{pmatrix} \sum_{a2=1}^{A2j} v(p_{j,a2}) - \sum_{i=1}^n \sum_{a1=1}^{A1i} vh(w_{i,a1}), \\ \sum_{b2=1}^{B2j} u(m_{j,b2}) - \sum_{i=1}^n \sum_{b1=1}^{B1i} uh(s_{i,b1}), \\ \sum_{c2=1}^{C2j} r(l_{j,c2}) - \sum_{i=1}^n \sum_{c1=1}^{C1i} th(e_{i,c1}) \end{pmatrix} \quad (14)$$

Далее для каждого вектора δ рассчитывается модуль $|\delta|$.

В итоге из нескольких ЦОДов выбирается тот, у которого $|\delta|$ принимает максимальное или минимальное значение.

B. Разделение пула запросов на подпулы для рассматриваемого ЦОД

Идея данного подхода деления пула запросов на подпулы заключается в следующем: для каждого из центров обработки данных алгоритму известны их остаточные ресурсы. Поступивший пул запросов

упорядочивается по возрастанию/убыванию рассматриваемой характеристики, затем отбираются заявки в подпул, пока их сумма по рассматриваемой характеристике не превысит остаточные ресурсы ЦОД.

Схема работы алгоритма:

- 1) Выбор k -ой характеристики;
- 2) Сортировка пула запросов по k -ой характеристике по возрастанию/убыванию;
- 3) Суммирование отсортированной последовательности до тех пор, пока сумма не превысит значение остаточного k -го ресурса;
- 4) Если сумма превысила пороговое значение, то отнимается значение текущей заявки из пула, запоминается номер предыдущей обозреваемой заявки;
- 5) Выбор следующей характеристики, переход к шагу 3 над полученным ранее пулом. Если все характеристики запроса просмотрены, то переход к шагу 6;
- 6) Передача полученного подпула локальному планировщику i -го ЦОДа. Если локальный планировщик не смог разместить подпул, то запросы из этого подпула возвращаются в исходный пул запросов.

Алгоритм завершает свою работу, когда все ЦОДы просмотрены.

C. Разделение пула поступивших запросов на подпулы для ЦОД, выбранного на основании критерия $|\delta|$

Среди доступных центров обработки данных для размещения поступившего пула запросов выбирается такой, у которого значение критерия $|\delta|$ принимает наибольшее или наименьшее значение.

Данный алгоритм является итерационным. Количество итераций выполнения алгоритма отслеживается с помощью счетчика итераций. Чтобы исключить случай бесконечного выполнения, необходимо ввести пороговое значение числа итераций, превышение которого позволит завершить работу алгоритма.

Пороговое значение равно $n * m$, где n – количество поступивших запросов, m – количество центров обработки данных.

Схема работы алгоритма:

- 1) Определение порогового значения;
- 2) Увеличение значения счетчика на единицу;
- 3) Составление подпула запросов, который мог бы быть размещен в i -м ЦОДе, по методу, описанному в пункте B;
- 4) Расчет критерия $|\delta|$ для i -го ЦОДа;
- 5) После того как для каждого ЦОДа получено значение критерия $|\delta|$, происходит поиск наименьшего/наибольшего значения этого критерия и соответствующий ему ЦОД.
- 6) Локальному планировщику найденного ЦОД происходит передача полученного подпула запросов. Если локальный планировщик не смог разместить подпул, то запросы из этого подпула возвращаются в исходный пул запросов.

- 7) Если во входном пуле запросов остались неразмещенные запросы, или если счетчик итераций не превысил пороговое значение, то переходим к шагу 2.

Алгоритм завершает свою работу, если все запросы из входного пула размещены, или если число попыток размещения запросов превысило пороговое значение.

IV. ЗАКЛЮЧЕНИЕ

В данной статье была сформулирована актуальность работы, приведена постановка задачи, описаны модели исходных данных и предложены различные способы построения алгоритмов мета-планирования ресурсов центров обработки данных.

БИБЛИОГРАФИЯ

- [1] Вдовин, П.М. Зотов И.А, Костенко В.А., Плакунов А.В., Смелянский Р.Л. Сравнение различных подходов к распределению ресурсов в центрах обработки данных // Известия РАН. Теория и системы управления, 2014., № 5.
- [2] Вдовин, П.М., Костенко В.А. Алгоритм распределения ресурсов в центрах обработки данных с отдельными планировщиками для различных типов ресурсов // Известия РАН. Теория и системы управления, 2014, № 6.

On research of various techniques of data center resource meta-scheduler algorithms

Ledovskiy M.V.

Abstract— This paper investigates the various techniques of constructing algorithms of data center resource meta-scheduler. It presents a description of a query model as well as a data center model. The mathematical problem of distributing requests across local planners of data center is given. The various techniques of the data center resource meta-scheduler algorithms are introduced.

Keywords— Cloud computing, data center, resource meta-scheduler.