УДК 004.272

В. В. Душеба, А. А. Сигарев, г. Киев

АЛГОРИТМ ВЫБОРА МАРШРУТА ТРАССИРОВКИ ИНФОРМАЦИОННОГО ПАКЕТА В HPC-СИСТЕМАХ

Abstract. In this article, the authors proposed the algorithm routing information packet trace in HPC-systems, which allows to simplify the communication protocols. This eliminates the need to refer to the operating system. Calculate the shortest route can compute node that generates an information package.

Введение. Современные HPC-системы (High Performance Computing) имеют своим основным назначением решение задач сверхбольших размерностей. При этом одна такая задача может полностью использовать все программно-аппаратные ресурсы системы [1-3]. Ранее авторами неоднократно отмечалось, что существует возможность полного упразднения межпроцессорного обмена в сетях между узлами, имеющими непосредственную связь [4, 5], а использование сетевых технологий обмена между такими узлами, как это делается сейчас, приводит к серьёзному падению производительности систем. Сетевые технологии необходимы для передачи информации только по маршруту на расстояние $\rho \geq 2$. Этот маршрут должен быть кратчайшим и единственным. На сегодня в сетях обычным является передача пакета через любой свободный узел, даже если при этом маршрут не оптимален. Такая методология может быть допустима в различных случаях, но в высокопроизводительных системах это приведет к увеличению среднестатистической протяженности маршрутов и падению реальной его производительности, особенно при решении единой распараллеленной задачи [6].

Об оптимизации использования сетевых технологий обмена.

Очевидно, что все фрагменты программы, имеющие сильную связь по данным должны загружаться в группы узлов, имеющих связь “каждого с каждым”. Чем слабее связь по данным между некоторой парой узлов, тем дальше они могут быть расположены в кластере. Такую упорядоченность необходимо осуществлять на уровне подготовки к решению задачи. Рассматриваемое семейство графов упрощает решение названных вопросов, т.к. позволяет априори вычислить кратчайшие маршруты для всех сообщений или пакетов системы и осуществлять оптимальную загрузку системы. Основная цель, которая при этом ставится, – максимальное упрощение процедуры межпроцессорного обмена и минимизация временных затрат на его осуществление.

Упрощение процедуры нахождения кратчайшего маршрута между парой любых произвольных вершин графа может послужить увеличению производительности системы.

Прежде чем перейти к поиску кратчайшего маршрута отметим одну закономерность – *кратчайший маршрут между двумя внутренними вершинами одной магистрали полностью пролегает по этой магистрали*.

Дальнейшие выводы проиллюстрируем на графе F_{16}^{3}. Данный граф имеет 16 полюсов и 120 магистралей. Вершинами данного графа является множество все трех разрядных 16-ричных чисел. Выберем произвольную магистраль, соединяющую полюса 999 и CCC, на которой зафиксируем маршрут, проходящий через вершины:

99C, 9C9, 9CC, C99, C9C,

и его двоичное отображение:

001, 010, 011, 100, 101.

Это отображение является упорядоченной последовательностью двоичных трех разрядных чисел (разность двоичных отображений смежных вершин равна 1).

Разность двоичных отображений конечной и начальной вершин маршрута определяет его протяженность:

$\rho = 101\text{-}001 = 100 = 4_{10}$.

Данный маршрут является кратчайшим, т.к. проходит по магистрали. Сравнивая структуру вершин, входящих в маршрут, и их двоичных отображений, легко заметить, что цифра “C” ведет себя, как “1” в двоичном отображении маршрута, а “9” соответственно - как “0”. Это говорит о том, что и структура вершин данного маршрута также является упорядоченной. Эта упорядоченность дает возможность сформулировать правило вычисления следующей вершины маршрута, вообще не анализируя инцедентных текущей вершине ребер.

Введем следующее обозначение. Цифру, которая ведет себя как “1”, будем называть псевдоединицей и маркировать нижним индексом *p*. Введём запись: $C \to C_p$, которая указывает на то, что псевдо единицей является цифра “C”, а цифра 9 будет являться псевдо нулём, который не будет специально маркироваться. Вводим операцию сложения псевдоединицы C_p с цифрой 9, которая будет выступать в качестве псевдонуля:

1. $C_p + 9 = C_p$.
2. $9C_p + C_p = C_p 9$.

Проведем подобное сложение с начальной вершиной маршрута

$99C_p + C_p = C_p 99$.

Результатом операции явилась следующая его вершина С99. Продолжив указанную операцию, можно просчитать весь маршрут. Может показаться, что это более чем тривиальный и вполне ожидаемый результат – заменили цифры 0 и 1 на другие два символа и получили то, что нам нужно. Если отвлечься от реального графа и от системы кодирования его вершин то, так оно на самом деле и будет, и единственный позитив будет состоять лишь в том, что операция проводилась над реальными вершинами. Отметим следующий момент: вершина 99С, в рассматриваемом примере, инцендентна 16 рёбрам и только одно из них входит в кратчайший маршрут.

Следовательно, проделанная операция без какого-либо просмотра других ребер выбрала именно ребро, входящее в кратчайший маршрут. Разумеется, что введенная операция сложения с псевдодвоичной единицей приобретает такие свойства только в данном типе графов и при данной системе кодирования их вершин.

Для реальных кластеров, построенных на предлагаемых принципах, это может означать возможность просчета кратчайших маршрутов передачи информации без использования громоздких протоколов обмена.

Определение кратчайших маршрутов передачи информации. Изложенные выше соображения позволяют рассматривать произвольное k-разрядное n-ричное число $A_0 = a_{k-1}, a_{k-2}, \ldots, a_0$, где $a_i \in M_n$, как позиционный псевдо двоичный код. При этом любая произвольная его цифра $b_{q-1} \in M_n$ может быть объявлена псевдо двоичной единицей, после чего все остальные цифры из M_n начинают выступать в функции псевдодвоичных нулей, но при этом разные цифры продолжают оставаться разными, но в двоичном отображении кода они заменяются нулями. Роль псевдоединицы в каждой конкретной ситуации может исполнять одна единственная цифра и только, она в двоичном отображении заменяется единицей. Теперь можно рассматривать операции с псевдодвоичными кодами, как операции с обычными двоичными числами, т.е.

$$\forall a_i \in M_n \left((a_i \neq b_{k-1}) \rightarrow (a_i + b_{k-1} = b_{k-1}) \right);$$

$$\forall a_j \in M_n \left((a_j \neq b_{k-1}) \rightarrow (a_j b_{k-1} b_{k-1} \ldots b_{k-1} + b_{k-1} = b_{k-1} a_j a_j \ldots a_j) \right).$$

Нами будет рассматриваться только операция сложения псевдо двоичного кода с псевдо единицей, которая будет использоваться для расчета кратчайших маршрутов между произвольной парой вершин графа. При этом любой символ псевдо двоичного кода может выступать в качестве псевдо единицы. Далее не будет особо маркироваться псевдо единица, а просто будет называться цифра, которая выступает в функции псевдо единицы.

Нахождение кратчайшего маршрута между двумя вершинами графа является одной из основных задач теории графов. Существует несколько основных алгоритмов её решения. Однако для графов вида F_n^k есть алгоритм, существенно отличный от известных, который базируется на специальной организации графа и особом методе кодирования его вершин.

Пусть имеются две произвольные вершины графа F_n^k : А и В, где

А= $a_{k-1}a_{k-2}\dots a_0 \in A_k$,

В= $b_{k-1}b_{k-2}\dots b_0 \in B_k$,

$A_k, B_k \supset M_n^k$

Вычислим кратчайший маршрут из А в В. Заметим, что вершина А $\supset$ $F_{n(a_{k-1})}^{k-1}$ и вершина В $\supset$ $F_{n(b_{k-1})}^{k-1}$. В качестве псевдо единицы назначаем b_{k-1} - старший разряд вершины В и начинаем складывать его со значением вершины А, все цифры которой будут выполнять функции псевдо нулей.

Рассмотрим случай, когда

$$A_k \cap B_k = 0. \qquad (1)$$

$$a_{k-1}a_{k-2}\dots a_0 + b_{k-1} = a_{k-1}a_{k-2}\dots b_{k-1}$$

$$a_{k-1}a_{k-2}\dots b_{k-1} + b_{k-1} = a_{k-1}a_{k-2}\dots b_{k-1}a_1$$

$$\vdots$$

$$a_{k-1}b_{k-1}\dots b_{k-1} + b_{k-1} = b_{k-1}a_{k-1}a_{k-1}\dots a_{k-1}$$

Как только b_{k-1} становится старшим разрядом в коде очередной точке маршрута, это будет означать, что маршрут вошёл и далее будет проходить в подграфе $F_{n(b_{k-1})}^{k-1}$. С этого момента в качестве псевдоединицы выбирается следующий по старшинству разряд b_{k-2}, которая начинает складываться с кодом последней полученной вершины маршрута:

$$b_{k-1}a_{k-1}\dots a_{k-1} + b_{k-2} = b_{k-1}a_{k-1}\dots b_{k-2}\,.$$

Далее, сложение продолжается до тех пока b_{k-2} не займет положенное ей место ($k-2$) разряда в результирующем коде, после чего псевдо единицей становится следующий разряд вершины В и т. д. до момента прихода в конечную точку маршрута – вершину В. Выполнение условия (1) указывает на то, что данный маршрут пролегает по одному из диаметров графа и его протяженность d= 2^k -1. Это следует из того, что двоичное отображение вершин этого маршрута требует k двоичных разрядов. Соблюдение условия (1) указывает на то, что псевдо единица не встречается в коде начальной вершины данного маршрута, и в этом случае его двоичное отображение будет равно $\underbrace{00\dots0}_{k}$, двоичное отображение его конечной его вершиной всегда будет $\underbrace{11\dots1}_{k}$. Это замечание имеет отношение только к маршрутам, протяженность которых равна диаметру. При несоблюдении условия (1) двоичные отображения начальных и конечных вершин могут иметь самые разные значения, не превосходящие диаметра графа, но процедура нахождения маршрута от этого не меняется.

Приведенная процедура нахождения кратчайшего маршрута является в предлагаемом методе типовой. Если протяженность маршрута по типовой процедуре $\rho < d$, то это ещё не означает, что данный маршрут является кратчайшим из всех возможных маршрутов. В рассматриваемом типе графов имеются контуры протяженность которых $L < 2d$. В том случае, если начальная и конечная точка маршрута находятся на таком контуре, то в конечную точку маршрута можно попасть с двух направлений, и оба маршрута могут быть по протяженности меньше диаметра. Одно из этих направлений обслуживается типовой процедурой маршрутизации, а другое направление требует особой процедуры, которая отличается от типовой нюансами.

Нетиповая процедура маршрутизации. Можно было бы на этом не останавливаться, но бывают случаи, когда именно нетиповая процедура приводит к кратчайшему маршруту.

Пример. Обратимся к графу F_{16}^{3}, в котором вычислим маршрут из точки 89C в точку C89. В соответствии с типовой процедурой имеем:

89C+C=8C9
8C9+C=8CC
8CC+C=C88
C88+9=C89

Как определить расстояние между рассматриваемыми точками, не вычисляя маршрута?

1. Вычисляем длину первого отрезка маршрута 89C $\rightarrow$ C88, на котором псевдо единицей является цифра C. Двоичное отображение этих двух вершин - соответственно 001 и 100. $\rho_1 = 100 - 001 = 11$, где ρ_1 -расстояние от 89C до C88.
2. Вычисляем второй отрезок маршрута C88 $\rightarrow$ C89, где псевдоединицей является цифра 9. Строим соответственные двоичные отображения 000 и 001. ρ_2 =001-000=1.
3. $\rho = \rho_1 + \rho_2 = 11 + 1 = 100 = 4_{10}$.

В общем случае в конечную точку можно попасть с n-1 возможного направления, обслуживаемого нетиповой процедурой маршрутизации, но в n-2 из них протяженность маршрута $l_{\min} > d$. По этой причине интерес могут представлять два направления.

Такой случай имеет место при невыполнении условия (1), когда в начальной и конечных вершинах равны (k-2) разряды.

Пример. Даны вершины: 1CC и 5CC. Найти кратчайший маршрут. Маршрут, найденный по типовой процедуре будет иметь протяженность $\rho = d = 7$. Условие (1) не выполняется, т. к. вершины отличаются только старшими разрядами. Проложим маршрут не по типовой процедуре через мост [15, 51], а через мост [1C, C1]. В этом случае в качестве псевдоединцы выбираем не старший разряд конечной вершины, а следующий по старшинсву разряд, которым является C и складываем его с кодом начальной вершины.

1CC+C=C11.

Начиная с этой вершины, далее пойдем в соответствии с типовой процедурой.

C11+5=C15, C15+5=C51, C51+5=C55, C55+C=5CC.

Получили $\rho = 5$. Таким образом, нетиповая процедура дает кратчайший маршрут. Таким образом, при нахождении кратчайшего маршрута в таких случаях необходимо принимать во внимание следующие по старшинству разряды кодов вершин.

В рассматриваемом типе графов всегда имеются контуры, вершины которых формируются посредством 3-х цифр, в которые не входят полюса графа, Назовем их троичными контурами. К примеру, контур 01, 10, 12, 21, 20, 02 и 01 в графе F_3^2. В графе F_4^2 имеется 4 троичных контура, число которых Q=C_n^3 (число сочетаний). Имеется в нём и контур подобный предыдущему, в который не вошла цифра 4. Протяженность троичного контура L находится из выражения (2):

$$L = 3 \cdot 2^{k-1}. \qquad (2)$$

Если пара точек, между которыми необходимо проложить маршрут, лежит на троичном контуре, расстояние между ними $\rho \le L/2 < d$.

Выводы. Априорное знание кратчайших маршрутов позволяет на этапе подготовки задачи распределить задачу по сети таким образом, чтобы слабо связанные её части загружались в самые отдаленные друг от друга узлы. Чем сильнее связь по данным между параллельными фрагментами программы, тем ближе они должны находиться в сети. Кроме того, пакет данных должен передаваться только по кратчайшему маршруту. Предлагаемый алгоритм позволяет максимально упростить протоколы обмена. При этом отпадает необходимость в обращении к операционной системе. Кратчайший маршрут может просчитывать вычислительный узел, который формирует информационный пакет.

1. *Loshin, D.* High Performance Computing Demystified, Cambridge, MA:AP Prof., 1994.
2. *Каляев И.А.* Реконфигурируемые мультиконвейерные вычислительные структуры / Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. // Изд. ЮНЦ РАН. – Ростов-на-Дону, 2008. – 393 с.

3. *Черняк Л.* Суперкомпьютинг вглубь и вширь / Леонид Черняк // Открытые системы, 2007. – № 09.
4. *Сигарев А.А., Душеба В.В.* Способ упразднения межпроцессорного обмена в макроконвейерах / А.А. Сигарев, В.В. Душеба // Электронное моделирование. – К., 2006. – т. 28, № 6. – С. 71-89.
5. *Сигарев А.А., Душеба В.В.* Организация многопроцессорных систем класса МКОД / А.А. Сигарев, В.В. Душеба // Электрон. моделирование. 1999. - 21, № 1. С. 47-57.
6. *Душеба В.В., Сигарев А.А.* К вопросу об оптимизации топологии кластерных вычислительных систем / В.В. Душеба, А.А. Сигарев // Моделювання та інформаційні технології : зб. наук. пр. – К.: ІПМЕ ім. Г.Є. Пухова НАНУ, 2011. – Вип. 61. – С. 32-36.

Поступила 19.02.2014р.

УДК 004.891.2

О. В. Маєвський, м.Тернопіль

ЗАДАЧА СТВОРЕННЯ ІНФОРМАЦІЙНОЇ БАЗИ МОНІТОРИНГУ ЗЕМНИХ І КОСМІЧНИХ ЯВИЩ

Abstract. In this paper the relevance of the information needs of users in the current state of environmental factors, including terrestrial and cosmic phenomena. To do this, the analysis of natural factors and the known systems receive data. The functional scheme of information system of mobile monitoring such factors.

Вступ

Дана робота присвячена задачам створення інформаційних баз моніторингу природніх факторів, зокрема, земних і космічних явищ з можливістю безперебійного доступу до даних пересічних громадян. Одними із найважливіших природніх факторів є сонячна активність, магнітне поле Землі, магнітна буря та як наслідок вплив магнітних бур на людей. В-першу чергу така інформація використовується в медичних організаціях, як при аналізі поточного стану здоров’я людей так і в задачах прогнозу захворювань, що дозволить передбачити можливі кризові ситуації.

Аналіз наукових праць по даній проблематиці підтвердив актуальність і важливість не тільки в галузі медицини.

Таким чином **метою** даної роботи є розробка інформаційної системи мобільного моніторингу природніх факторів, доступної пересічному користувачеві.

Для досягнення поставленої мети необхідно вирішити ряд наступних завдань:

1. Проаналізувати природні фактори, їх основні характеристики та діючі системи і джерела отримання такої інформації;