Приближенный алгоритм выбора оптимального подмножества узлов в коммуникационной сети Ангара с отказами

А. В. Мукосей, А. С. Семенов, Д. В. Макагон
АО «НИЦЭВТ»

В АО «НИЦЭВТ» разрабатывается высокоскоростная коммуникационная сеть Ангара с топологией «многомерный тор». При реальном использовании суперкомпьютера с сетью Ангара в условиях наличия занятых и отказавших узлов возникает задача нахождения оптимального подмножества узлов сети для покрытия заданного числа узлов так, чтобы весь сетевой трафик лежал только внутри этого подмножества узлов. В данной работе представлен приближенный полиномиальный алгоритм решения такой задачи.

Ключевые слова: Отказоустойчивость, коммуникационные сети, многомерный тор, связность, детерминированная маршрутизация, маршрутизация с порядком направлений.

1. Введение

В АО «НИЦЭВТ» разрабатывается высокоскоростная коммуникационная сеть Ангара [1, 2] с топологией «многомерный тор». В маршрутизаторе сети реализована бездедлоковая, адаптивная маршрутизация, основанная на правилах «пузырька» (Bubble flow control, [4]) и «порядка направлений» (Direction ordered routing, DOR, [5, 6]) с использованием битов направлений [6]. Благодаря алгоритму First Step/Last Step «нестандартного первого и последнего шага» [6] аппаратно поддерживается обход отказавших узлов или линков. Эффективность этого метода по поддержанию связности в сети с отказами была показана в статье [3]. Применяемая маршрутизация позволяет избежать взаимных блокировок из-за циклических зависимостей пакетов в кольцах и между кольцами нескольких измерений, а также гарантирует сохранение порядка передачи пакетов между любыми двумя адресатами.

Для эффективного использования узлов системы, необходимо уметь оптимально выделять ресурсы в зависимости от состояния кластера. Состояние кластера изменяется постоянно по различным причинам: занятость отдельных узлов и неисправность оборудования. Необходимо уметь за небольшое время принимать решения по выделению требуемого числа узлов. В данной статье предложен приближенный алгоритм выбора оптимального подмножества узлов в коммуникационной сети Ангара с отказами. Авторам статьи подобные алгоритмы для сетей с топологией «многомерный тор» и используемыми в сети Ангара методами маршрутизации в литературе неизвестны.

Статья организована следующим образом. В разделе 2 приводятся необходимые формальные определения. В разделе 3 описывается маршрутизация в сети Ангара. В разделе 4 представлена постановка задачи. В разделе 5 рассматриваются алгоритмы решения поставленной задачи: вспомогательный алгоритм определения существования маршрутов из каждого узла множества в другой каждый узел этого множества, алгоритм выбора множеств узлов равномерным расширением, приближенный алгоритм расчета оптимальных таблиц маршрутизации для заданного множества узлов, также в этом разделе описан используемый в настоящее время в сети Ангара алгоритм выбора требуемого множества узлов без учета занятых или отказавших узлов. Исследование разработанных алгоритмов проводится в разделе 6.

2. Определения

В данном разделе вводятся некоторые формальные определения, которые в дальнейшем будут использоваться в статье.

Рассмотрим коммуникационную сеть с топологией многомерный тор. Множество всех узлов сети обозначим N, размерности тора обозначим $(d_1, d_2, ..., d_n)$, а общее число узлов — $|N|$. Каждый узел u имеет координаты $(u_1, u_2, ..., u_n)$, где $0 \le u_i < d_i$. Соседними в рамках тороидальной топологии будем называть узлы $u = (u_1, u_2, ..., u_n)$ и $\tilde{u} = (u_1, ..., (u_j \pm 1)\ mod\ d_j, ..., u_n)$ для любого индекса $1 \le j \le n$.

Легко заметить, что каждый узел имеет $2n$ соседей (в случае $d_j > 2$). Будем считать, что каждый из этих соседей находится в одном из *направлений* от узла u.

Множество направлений обозначим $\mathcal{D}$ и пронумеруем их числами $\overline{1, 2n}$:

$$\mathcal{D} = \{\triangle_j\}_{j=\overline{1,2n}}.$$

Направления с номерами $1, ..., n$ будем называть *положительными*:

$$\tilde{u} = (u_1, ..., (u_j + 1)\ mod\ d_j, ..., u_n),$$
$$\triangle_j = (0, ..., \underbrace{1}_{\text{позиция } j}, ..., 0) \in \mathcal{D},$$

где $1 \le j \le n$.

Направления с номерами $n+1, ..., 2n$ будем называть *отрицательными*:

$$\tilde{u} = (u_1, ..., (u_{j-n} - 1)\ mod\ d_{j-n}, ..., u_n),$$
$$\triangle_j = (0, ..., \underbrace{-1}_{\text{позиция } j-n}, ..., 0) \in \mathcal{D},$$

где $n + 1 \le j \le 2n$. В такой формулировке выражение «узел u находится в направлении D от узла v записывается как $u = v + D$».

На множестве направлений $\mathcal{D}$ введем порядок в соответствии с указанной нумерацией: $\triangle_i < \triangle_j$, если $i < j$.

Определение 1. Каналом связи (линком) будем называть пару (u, D), где $u \in N$, $D \in \mathcal{D}$. Множество всех каналов связи обозначим $\mathcal{E} = N \times \mathcal{D}$.

Определение 2. Путем $\mathcal{P}$, соединяющим два узла сети: u^0 и u^n, назовем последовательность вида $u^0, D_1, u^1, D_2, ..., D_N, u^N$, где u_i — узел сети, D_i — направления, связывающее узел u^{i-1} с узлом u^i, N — длина пути. При этом $u^1, ..., u^{n-1}$ — *транзитные* узлы. Так как транзитные узлы могут быть получены из соответствующих переходов, то их можно опустить, тогда подобный путь будет записываться в виде: $u^0, D_1, D_2, ..., D_n$.

Определение 3. Подмножество M множества узлов N назовем маршрутизируемым, если для любых двух узлов u, v множества M существует путь $\mathcal{P}$ из u в v такой, что транзитные узлы этого пути принадлежат M.

Определение 4. Таблицей маршрутизации $\mathcal{R}$ маршрутизируемого множества M назовем некоторый набор путей таких, что для любых двух узлов u, v множества M в $\mathcal{R}$ существует единственный путь из u в v такой, что транзитные узлы этого пути принадлежат M.

Определение 5. Диаметром маршрутизируемого множества M с таблицей маршрутизации $\mathcal{R}$ назовем максимальную длину пути из набора путей $\mathcal{R}$.

Определение 6. Пусть для некоторого маршрутизируемого множества $M \subset N$ построена таблица маршрутизации $\mathcal{R}$. Загруженностью $G_{(u,D)}$ канала связи маршрутизируемого множества M будем называть количество путей, которым принадлежит данный канал связи: $G_{(u,D)} = |\ \{\mathcal{P}_{ij} \mid (u, D) \in \mathcal{P}_{ij}, \mathcal{P}_{ij} \in \mathcal{R}\}\ |$.

3. Маршрутизация в сети Ангара

3.1. Правило порядка направлений с использование битов направлений

Среди алгоритмов маршрутизации для многомерных торов можно выделить класс алгоритмов, соблюдающих *правило порядка направлений*: маршрут между любой парой узлов включает движения в направлениях в определенном, заранее заданном, порядке. Эти алгоритмы обладают свойством отсутствия взаимных блокировок между кольцами нескольких измерений тора при любом количестве одновременных запросов на передачу данных по сети.

Во введенных обозначениях правило порядка направлений будет формулироваться следующим образом: $D_{j-1} \leq D_j, j = \overline{2, N}$, где N — *длина* пути.

Чтобы задать путь, удовлетворяющий правилу порядка направлений, необходимо задать стартовую вершину и количество шагов в каждом из направлений, т.е. набор $u^0, s_{\delta(1)}, s_{\delta(2)}, ..., s_{\delta(i)}$, где $u^0 \in N$ — стартовый узел, $s_{\delta(1)}, s_{\delta(2)}, ..., s_{\delta(i)} > 0$ — количество шагов в направлениях $\triangle_{\delta(1)}, \triangle_{\delta(2)}, ..., \triangle_{\delta(i)}$ таких, что $\triangle_{\delta(j)} < \triangle_{\delta(j+1)}$, $j = \overline{1, i-1}$.

В сети Ангара реализована маршрутизация с использованием *битов направлений*, которая вносит некоторые ограничения на маршрутизацию с правилом порядка направлений. Аналогично правилу порядка направлений для задания пути, соответствующему маршрутизации с использованием битов направлений, необходимо задать стартовую вершину и количество шагов в выбранных направлениях, то есть следующий набор: $u^0, s_{\delta(1)}, s_{\delta(2)}, ..., s_{\delta(i)}$, где $u^0 \in N$ – стартовый узел, $s_{\delta(1)}, s_{\delta(2)}, ..., s_{\delta(i)} > 0$ – количество шагов в направлениях $\triangle_{\delta(1)}, \triangle_{\delta(2)}, ..., \triangle_{\delta(i)}$. При этом в наборе направлений $\triangle_{\delta(1)}, \triangle_{\delta(2)}, ..., \triangle_{\delta(i)}$ нет направлений с противоположными знаками и $\triangle_{\delta(j)} < \triangle_{\delta(j+1)}$, $j = \overline{1, i-1}$. Обозначим такой набор направлений как D_{dirbit}. Путь, соответствующий маршрутизации с использованием битов направлений, обозначим P_{dirbit}.

3.2. First Step/Last Step

Метод First Step/Last Step [6] используется в сети Ангара как механизм обхода отказавших узлов. Он расширяет маршрутизацию с использованием битов направлений путем добавления первого и последнего нестандартного шага.

Путь с использованием первого и последнего нестандартного шага будет записываться следующим образом: $u^0, D_{FS}, P_{dirbit}, D_{LS}$, где u^0 — стартовый узел, D_{FS} — первое положительное нестандартное направление, D_{LS} — последнее отрицательное нестандартное направление. При этом набор направлений $D_{FS}, D_{dirbit}, D_{LS}$ удовлетворяет правилу порядка направлений.

Таким образом, для однозначного задания пути в сети Ангара необходимо задать набор $D_{FS}, P_{dirbit}, D_{LS}$.

4. Постановка задачи

Во время работы разделяемого вычислительного кластера необходимо при любом состоянии системы уметь предоставлять требуемое число узлов, которые должны быть маршрутизируемы между собой и не иметь транзитного трафика вне этого набора узлов, если это возможно. Состояние системы определяется набором отказавших линков и/или узлов и наличием занятых узлов. Занятый или отказавший узел можно интерпретировать как узел, у которого линки сломаны во всех направлениях.

Обозначим множество сломанных линков $F \subset \mathcal{E}$.

Так как физический канал связи между двумя узлами v и u представляет собой линки от узла v к узлу u и наоборот, то разумно предположить, что при неисправности одного из линков — второй так же неисправен. Таким образом, множество F будет включать в себя

отказавшие каналы связи попарно.

Во введенных определениях задача будет формулироваться следующим образом. Пусть задан тор с размерностями $(d_1, ..., d_n)$ и набором отказавших линков F. Требуется построить алгоритм нахождения маршрутизируемого множества M такого, что $m \leq |M|$, где m — требуемое число узлов.

Так как различных систем M может быть несколько, необходим критерий выбора оптимального маршрутизируемого множества. В работе рассматривались следующие критерии:

1. Минимальный диаметр;
2. Наименьшая средняя загрузка линков;
3. Наименьшее число транзитных узлов.

Первый критерий возникает из-за того, что в сети с минимальным диаметром задержка на передачу данных будет наименьшей. Второй критерий следует из стремления получить равномерно загруженную систему. Третий критерий — из необходимости эффективно использовать аппаратные ресурсы вычислительного кластера.

5. Алгоритмы решения задачи

Для решения поставленной задачи разработано несколько алгоритмов. Основной алгоритм выбора множеств узлов равномерным расширением (см. подраздел 5.2) приведен после используемого им вспомогательного алгоритма проверки множества на маршрутизируемость (см. подраздел 5.1). В алгоритме равномерного расширения строится набор множеств требуемого размера, после чего требуется выбрать оптимальное множество. Приближенный алгоритм расчета критериев оптимальности и выбора множества, которое является решением задачи, приведен в подразделе 5.3.

Для оценки качества предложенного алгоритма в подразделе 5.4 приведен используемый в настоящее время в сети Ангара алгоритм выбора требуемого множества узлов без учета занятых или отказавших узлов.

5.1. Алгоритм определения маршрутизируемости множества

Сведем задачу определения маршрутизируемости множества к поиску пути в некотором ориентированном графе $G(V, E)$.

При построении маршрута между двумя узлами ограничение на принятие решения о следующем шаге вносит предыстория пути. Рассмотрим движение по некоторому пути в торе в направлениях: $\triangle_{\delta(1)}, ..., \triangle_{\delta(i)}$. Этот путь можно продолжить только в таком направлении $\triangle_{\delta(i+1)}$, что набор направлений $\triangle_{\delta(0)}, ..., \triangle_{\delta(i)}, \triangle_{\delta(i+1)}$ удовлетворяет правилу с использованием битов направлений или $\triangle_{\delta(i)} \leq \triangle_{\delta(i+1)}$ в случае, если $\triangle_{\delta(i+1)}$ является последним нестандартным шагом.

Поэтому для описания вычислительного узла u^i в графе построим множество U^i вершин, которые будут характеризовать предысторию путей, которые проходят через вычислительный узел u^i:

1. $U^i_{D_{FS_j}}, j = 1, .., n$ — вершины, в которые возможно попасть, совершив первый нестандартный положительный шаг из соседнего узла в направлении D_{FS_j};
2. $U^i_{D_{LS_j}}, j = n+1, .., 2n$ — вершины, в которые возможно попасть, совершив последний нестандартный отрицательный шаг из соседнего узла в направлении D_{FS_j};
3. $U^i_{D_{dirbit_j}}$, $D_{dirbit_j} \in D_{dirbit}$ — всевозможные наборы направлений, удовлетворяющие правилу с использованием битов направлений, за исключением набора с отсутствием

движения. В эти вершины возможно попасть, совершив движение по соответствующему набору направлений;

4. U^i_{begin} — вершина, из которой начинается движение.

5. U^i_{end} — вершина, в которой заканчивается движение.

Посчитать количество вершин $U^i_{D_{dirbit_j}}$ можно следующим образом. Закодируем набор n-мерным числом, где на j-ом месте стоит -1 в случае движения в направлении $\triangle_{j+n}$, $+1$ в случае движения в направлении $\triangle_j$ и 0 — в случае отсутствия движения в j-том измерении тора. Всего таких наборов $3^n - 1$.

Таким образом,

$$U^i = (\cup_{j=1}^{n} U^i_{D_{FS_j}}) \cup (\cup_{j=n+1}^{2n} U^i_{D_{LS_j}}) \cup (\cup_{j=1}^{3^n-1} U^i_{D_{dirbit_i}}) \cup U^i_{begin} \cup U^i_{end},$$
$$|U^i| = n + n + 3^n - 1 + 2 = 3^n + 2n + 1,$$
$$V = \cup_{i=1}^{|N|} U^i,$$
$$|V| = |N| * |U^i| = |N| * (3^n + 2n + 1) = A|N|.$$

Соединим вершины в графе следующим образом.

1. Рассмотрим вершину U^i_{begin}, соответствующую узлу u^i, из которой начинается движение. Первым шагом в пути из узла u^i может быть:

   - движение в направлении первого нестандартного положительного шага D_{FS_k} в вершину $U^j_{D_{FS_k}}$, соответствующую узлу $u^j = u^i + D_{FS_k}$,
   - движение в любом направлении $\triangle_k$ в вершину $U^j_{\triangle_k}$, соответствующую узлу $u^j = u^i + \triangle_k$,
   - в вершину U^i_{end} для завершения движения.

2. Рассмотрим вершины $U^i_{D_{FS_l}}$, соответствующие узлу u^i. Из этих вершин возможно движение в вершины $U^j_{\triangle_k}$, соответствующие узлам $u^j = u^i + \triangle_k$ в направлениях $\triangle_k$ таких, что $D_{FS_l} < \triangle_k$.

3. Рассмотрим вершины $U^i_{\triangle_{\delta(0)},...,\triangle_{\delta(l)}}$, соответствующие узлу u^i, через который проходят пути с направлениями $\triangle_{\delta(0)}, ..., \triangle_{\delta(l)}$. Из этих вершин возможно движение:

   - в вершины $U^j_{\triangle_{\delta(0)},...,\triangle_{\delta(l)},\triangle_{\delta(l+1)}}$, соответствующие узлам $u^j = u^i + \triangle_{\delta(l+1)}$ в направлениях $\triangle_{\delta(l+1)}$ таких, что $\triangle_{\delta(0)}, ..., \triangle_{\delta(l)}, \triangle_{\delta(l+1)}$ удовлетворяет правилу с использованием битов направлений,
   - в вершины $U^j_{D_{FS_k}}$, соответствующие узлам $u^j = u^i + D_{FS_k}$ в направлениях D_{FS_k} таких, что $\triangle_{\delta(0)}, ..., \triangle_{\delta(l)}, D_{FS_k}$ удовлетворяет правилу порядка направлений,
   - в вершину U^i_{end} для завершения движения.

4. Рассмотрим вершины $U^i_{D_{LS_k}}$, соответствующие узлу u^i, через который проходят пути с последними нестандартными отрицательными направлениями D_{LS_k}. Из этих вершин возможно движение только в вершину U^i_{end} для завершения движения.

На рисунке 1 изображены все вершины графа, соответствующие узлу u^i в двухмерной топологии сети. Узлы u^a, u^b, u^c, u^d — соседние узлы к узлу u^i. Пунктиром обведены вершины графа, которые соответствуют одному узлу сети. На рисунке изображенны только ребра, входящие в вершины графа, которые относятся к узлу u^i. . Основной алгоритм

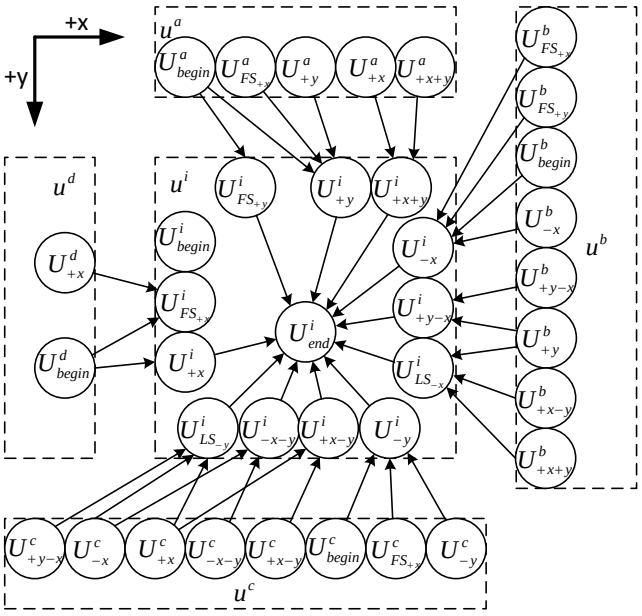


Рис. 1. Часть графа на двухмерной топологии.

Посчитаем, сколько ребер графа приходится на один набор U^i вершин. В первом случае вершины U^i_{begin} имеют $n + 2n + 1$ ребер. Во втором случае для каждого D_{FS_j} существует $2n - j$ вариантов, итого получим $\sum\limits_{j=1}^{n}(2n - j) = 2n * n - \frac{(1+n)n}{2} = \frac{3n^2-n}{2}$ ребер. В третьем случае для каждой вершины $U^i_{\triangle\delta(0),...,\triangle\delta j}$ из $3^n - 1$ вершин будет не больше, чем $2n$ соседей. В четвертом случае — n ребер.

Просуммировав все, получим $n+2n+1+\frac{3n^2-n}{2}+(3^n-1)2n+n = 2n3^n+1.5n^2+1.5n+1 = B$ — оценку числа ребер на каждый узел тора u^i.

Таким образом, общее количество ребер в графе $|E| = O(B|N|)$.

Утверждение 1. Из узла $u^i \in N$ в узел $u^j \in N$ существует путь $\mathcal{P}$ тогда и только тогда, когда в графе $G(V, E)$ существует путь из вершины U^i_{begin} в вершину U^j_{end}.

Доказательство 1. Для доказательства приведем взаимнооднозначное соответствие между множеством путей в сети и множеством путей в графе G. Рассмотрим путь:

$$P = u^i, D_{FS}, \underbrace{\triangle_{\delta(1)}, ..., \triangle_{\delta(1)}}_{s_{\delta(1)}>0}, ..., \underbrace{\triangle_{\delta(i)}, ..., \triangle_{\delta(i)}}_{s_{\delta(i)}>0}, D_{LS} = v^j$$

Построим соответствующий путь в графе G:

$$U^i_{begin}, U^1_{FS}, \underbrace{U^2_{\triangle_{\delta(1)}}, ..., U^{2+s_{\delta(1)}}_{\triangle_{\delta(1)}}}_{s_{\delta(1)}}, ..., \underbrace{U^{\sum_{j=1}^{i-1} s_{\delta(j)+2}}_{\triangle_{\delta(1)},...,\triangle_{\delta(i)}}, ..., U^{\sum_{j=1}^{i} s_{\delta(j)+2}}_{\triangle_{\delta(1)},...,\triangle_{\delta(i)}}}_{s_{\delta(i)}}, U^j_{LS}, U^j_{end}$$

Аналогичным образом по данному пути в графе G можно построить путь в сети, удовлетворяющий правилам маршрутизации. Это соответствие показывает, что существование пути в сети из u^i в u^j равносильно существованию пути в графе G из U^i_{begin} в U^j_{end}.■

Таким образом, задача определения маршрутизируемости множества $M \subset N$ сводится к определению связности множества вершин в графе G, соответствующих узлам множества

M. Для этого можно применить метод поиска вширь: для каждой из вершин U^i_{begin} определим множество достижимых из нее вершин вида U^j_{end}. При этом соответствующие узлы u^j будут достижимы из u^i, а u^j, u^i и транзитные узлы будут принадлежать M.

Алгоритм поиска вширь имеет сложность $T = O(V + E) = O(A|M| + B|M|) = O((A + B)|M|)$. Так как алгоритм нужно выполнить для всех M вершин, то $T = O((A + B)|M|^2)$.

5.2. Алгоритм выбора подмножеств узлов равномерным расширением

Рассмотрим переборный алгоритм решения задачи выбора оптимального подмножества узлов в сети с отказами. Всего вариантов расположения m узлов в сети: $\binom{m}{Nnodes}$. Для проверки маршрутизируемости для каждого набора узлов требуется m^2 операций. Даже для небольших систем $\binom{m}{Nnodes} * m^2$ очень велико, эта функция растет очень быстро, поэтому переборный алгоритм не может быть применен на практике.

В качесте решения задачи предлагается приближенный алгоритм решения задачи полиномиальной сложности. Идея алгоритма заключается в том, что из каждого узла тора происходит попытка построить k-мерный прямоугольник поочередным (равномерным) расширением в разные стороны. Ниже алгоритм будет называться алгоритмом равномерного расширения.

На вход алгоритму подается размер искомой системы m. Алгоритм состоит из трех этапов: на первом из каждого узла сети производится расширение во все стороны с добавлением только тех узлов, у которых нет отказавших линков. На втором этапе проводится сортировка полученных систем и удаление одинаковых. На третьем этапе производится расширение получившихся систем с добавлением отказавших линков.

Рассмотрим алгоритм подробнее. На первом этапе алгоритм выделяет прямоугольники без сломанных линков. Такие прямоугольники хороши тем, что не требуют проверки множества на маршрутизируемость. Для каждого узла u^i сети строится многомерный прямоугольник следующим образом. Каждый прямоугольник можно задать двумя узлами в противоположных углах: Pa и Na. Первоначальный прямоугольник состоит из одной вершины: $Pa = Na = u^i$. Затем происходят попытки увеличения прямоугольника в каждом направлении по очереди с проверкой, что в захваченной области нет сломанных линков. Расширение прямоугольника происходит путем перемещения одного из его углов в выбранном направлении — Pa в случае расширения в положительном направлении, Na в случае отрицательного. После каждого расширения происходит проверка числа захваченных узлов, если это число больше m, то первый этап завершается. Также первый этап завершается, если отсутствуют направления, в которые можно расширить прямоугольник. На рисунке 2 представлена схема работы первого этапа алгоритма для двухмерного случая.

Оценим сложность первого этапа алгоритма. За все время обхода нужно проверить $O(2n|M| + |M|) = O((2n + 1)|M|)$ узлов и линков. Этот алгоритм нужно применить ко всем узлам множества N. Получаем следующую оценку сложности: $T_{s_1} = O((2n + 1)|M||N|)$.

Для того, чтобы сократить дальнейшую работу, на втором этапе происходит удаление одинаковых построенных прямоугольников при помощи предварительной сортировки. Если на втором этапе находятся прямоугольники нужного размера, алгоритм заканчивается. Так как каждый прямоугольник задается с помощью двух узлов сети, то его можно описать набором $2n$ чисел. Таким образом, чтобы сравнить все прямоугольники, необходимо $O(|N|log_2(|N|))$ сравнений. Таким образом, оценка сложности второго этапа $T_{s_2} = O(|N|log_2(|N|))$.

На третьем этапе происходят попытки расширить полученные прямоугольники в стороны со сломанными линками, на каждом шаге проверяется маршрутизируемость результирующего множества узлов. Заметим, чтобы проверить маршрутизируемость нового множества M'', которое получится из M путем расширения в выбранном направлении, необходимо проверить маршрутизируемость добавленных узлов M' со всеми узлами M''. Это можно сделать с помощью построенного графа $G(V, E)$. Для этого из каждой вершины множества

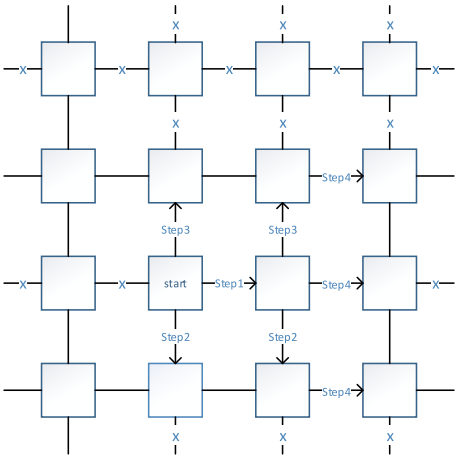


Рис. 2. Схема работы приближенного алгоритма равномерного расширения на примере двухмерного тора.

M' можно пройти поиском вширь по графу $G(V,E)$ и графу $G^T(V,E)$, полученному из графа $G(E,V)$ путем обращения связей (стартовая вершина теперь будет U^j_{end}, а конечная — U^i_{begin}). Таким образом, для каждой вершины u^i из M' можно получить множество узлов u^j, для которых существует путь в одну сторону и обратно.

Сложность третьего этапа в наихудшем случае можно оценить случаем, когда во всех расширениях прямоугольника присутствовали сломанные линки, а значит для всех узлов множества M пришлось выполнить поиск в ширь по графу $G(V,E)$ и графу $G^T(V,E)$ – $T_{s_3} = O(2(A+B)|M|^2|N|)$.

Алгоритм равномерного расширения можно оценить как $T_{expan} = T_{s_1} + T_{s_2} + T_{s_3} = O((2n+1)|M||N| + |N|log_2|N| + 2(A+B)|M|^2|N|) = O((A+B)|M|^2|N|)$, где $A = 3^n + 2n + 1$ и $B = 2n3^n + 1.5n^2 + 1.5n + 1$, $A + B = (2n+1)3^n + 1.5n^2 + 3.5n + 2$.

Значение констант A и B довольно велико, для сети размерностью $n = 4$ значение выражения $A + B = 769$. Однако $A + B \leq |N|$, поэтому предложенный алгоритм является полиномиальным.

В результате работы алгоритма получается набор маршрутизируемых множеств размера больше или равного m. Затем необходимо выбрать оптимальное множество. Для этого необходимо вычислить значение характеристик: диаметра, средней загрузки линков и число транзитных узлов системы и выбрать наилучшее множество путем сортировки сначала по диаметру, затем по средней загруженности и затем по числу транзитных узлов (см. критерии выбора в разделе 4 постановки задачи). Алгоритм вычисления средней загруженности линков системы путем построения таблиц маршрутизации представлен в следующем подразделе.

5.3. Алгоритм построения таблиц маршрутизации

Пусть имеется маршрутизируемое множество $M \subset N$. Требуется найти оптимальную таблицу маршрутизации для узлов множества M и вычислить загруженность каждого линка всех таких узлов M.

Допустим, что число путей между двумя узлами ограничено некоторым числом N_{paths}, тогда существует $N_{paths}^{(|M|-1)*|M|}$ различных таблиц маршрутизаций. Даже при небольшом числе узлов сети и различных вариантов путей число различных таблиц маршрутизации очень велико, и требуется специальный алгоритм для создания таблиц маршрутизации.

Предложен следующий алгоритм построения таблицы маршрутизации. Предположим, что все линки узлов множества M имеют нулевую загруженность. Для каждого узла u

маршрутизируемого множества M в графе $G(V, E)$ запускается алгоритм поиска вширь. После окончания поиска из каждого узла множества M необходимо подняться по построенному дереву обратно вверх к узлу u, увеличивая при этом загруженность $G_{u,D}$ проходимых линков сети. Эвристически выяснено, что сбалансированная таблица маршрутизации получается, если в качестве следующего узла для запуска поиска вширь выбирать максимально удаленным от узла u. Вторая эвристика, введенная для получения более равномерной загрузки линков, заключается в сортировке вершин на каждом новом слое поиска вширь по возрастанию загруженности линков, соответствующих вершинам.

Сортировку слоев в алгоритме можно оценить как $O(\sum_{i=1}^{l}(|V_i|log_2|V_i|)) =$

$= O(\sum_{i=1}^{l}(|V_i|log_2|V|)) = O(|V|log_2|V|)$, где l — число слоев в алгоритме, V_i — множество вершин на каждом слое.

Сложность одного прохода этого алгоритма можно оценить как сумму трех слагаемых: $T_1 = O((A + B)|M|^2)$ для поиска вширь в графе $G(V, E)$, $T_2 = O(|V|log_2|V|)$ — сортировка узлов на каждом шаге поиска, $T_3 = O(L_{max}|M|)$ — вычисление загруженности линков.

В худшем случае таблицы маршрутизации нужно построить для каждой системы, построенной алгоритмом равномерного расширения из каждого узла сети. Поэтому итоговая сложность постройки таблиц маршрутизации для всех систем составляет $T_R = |N| * (T_1 + T_2 + T_3) = O((A + B)|M|^2|N|)$. Заметим, что алгоритм построения таблиц маршрутизации имеет такую же сложность, как и алгоритм равномерного расширения.

5.4. Алгоритм решения задачи в сети без отказов

Для того, чтобы оценивать качество работы разработанного приближенного алгоритма равномерного расширения, проводилось сравнение с алгоритмом решения задачи выбора оптимального маршрутизируемого множества в сети без отказов, работающего по принципу факторизации.

Алгоритм выбора оптимального множества узлов в сети без отказов устроен следующим образом. Для системы размера m строятся всевозможные разложения чисел $m, m+1, ..., |N|$ на n натуральных множителей $k_1, ..., k_n$ таких, что $\forall i,\ k_i \leq d_i$, которые характеризуют прямоугольную область. Эта область является одним из решений задачи. Для всех таких решений ищется система с минимальным диаметром. Дополнительно рассчитывается загруженность линков при помощи алгоритма построения таблиц маршрутизации в подразделе 5.3, а также количество транзитных узлов.

Алгоритм факторизации решения задачи в сети без отказов используется в данный момент в системном ПО на кластере с сетью Ангара.

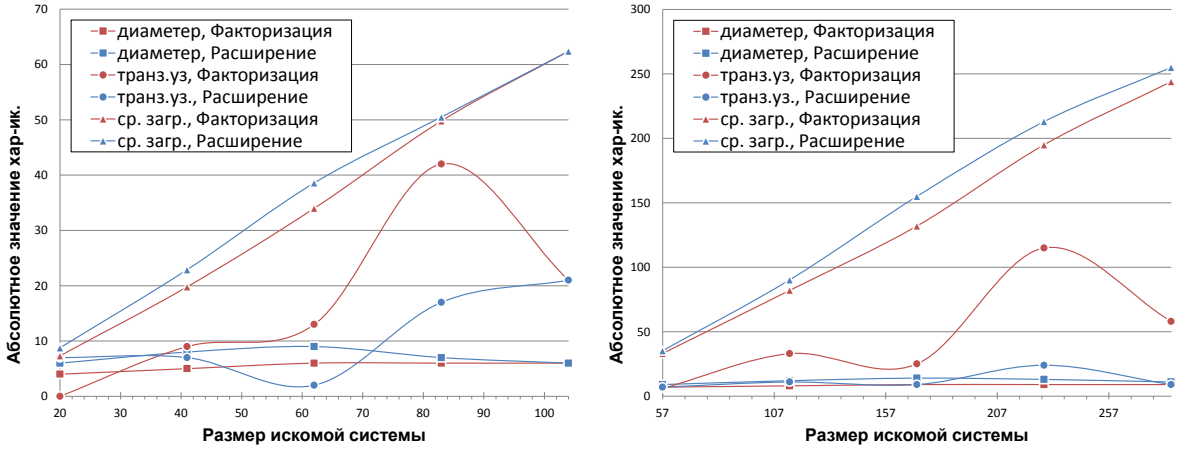
6. Исследование

Исследование качества разработанного приближенного алгоритма выбора оптимального подмножества узлов в сети с отказами можно провести следующим образом. Сначала с помощью сравнения результатов работы разработанного алгоритма с алгоритмом выбора узлов в сети без отказов проводится оценка качества нового алгоритма в том случае, для которого известен другой алгоритм решения задачи. Затем проводится исследование результатов работы нового алгоритма в сети с отказами в зависимости от размера искомой системы и количества сломанных линков.

Исследование разработанного алгоритма проводилось на равносторонних трехмерных торах: 5x5x5, 7x7x7 и 9x9x9.

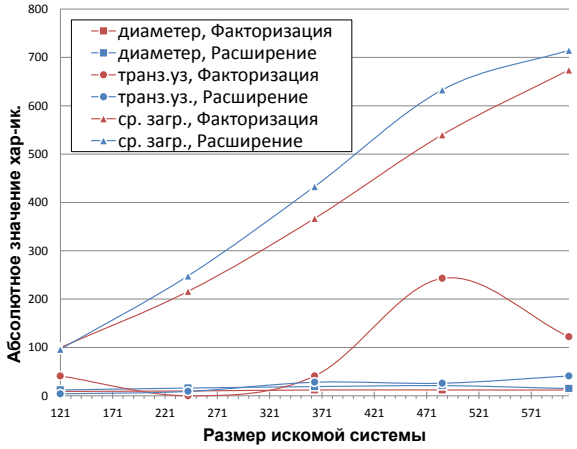
На рисунке 3 представлены характеристики систем, полученных с помощью алгоритма выбора подмножества узлов в сети без отказов (алгоритм факторизации) и алгоритма равномерного расширения в зависимости от размера сети и размера искомой системы. По

диаметру и средней загруженности линка на алгоритме равномерного расширения получены значения, немного уступающии алгоритму факторизации, а число транзитных узлов для алгоритма равномерного расширения значительно лучше. Так как алгоритм равномерного расширения более общий по сравнению с алгоритмом факторизации, то полученные незначительные ухудшения характеристик являются допустимыми.



(a) Топология сети 5x5x5

(b) Топология сети 7x7x7

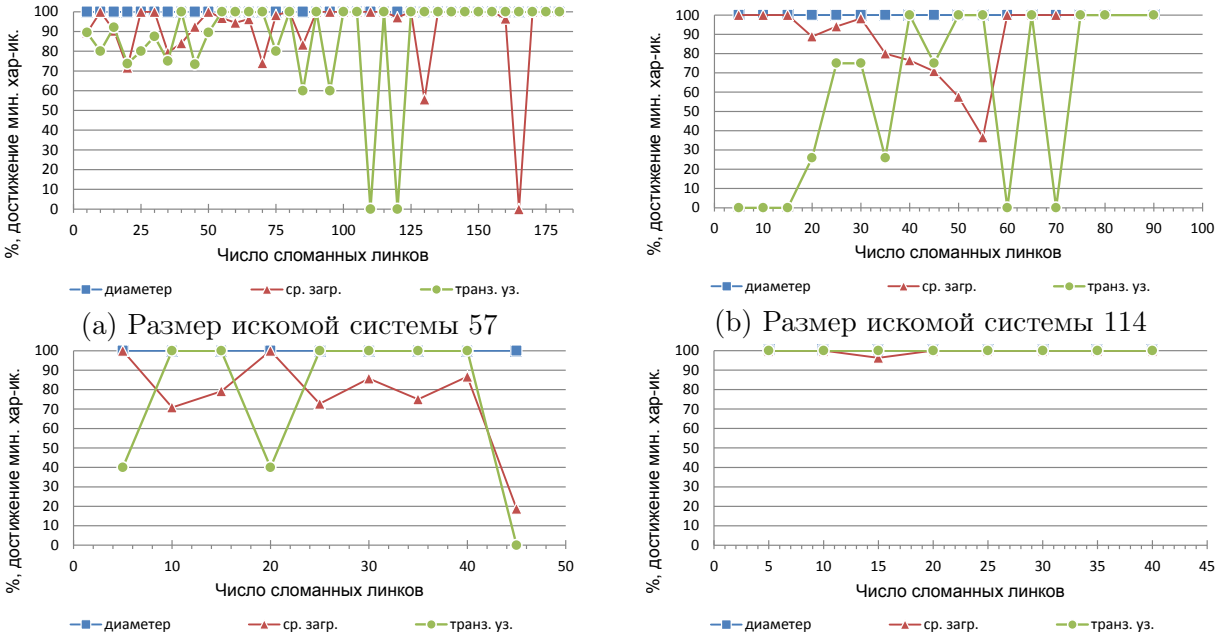


(c) Топология сети 9x9x9

Рис. 3. Значение характеристик решения задачи в зависимости от размера искомой системы.

Исследование алгоритма равномерного расширения проводилось в сети с топологией 7x7x7 (число узлов $N_{nodes} = 343$) с отказами. Сломанные линки выбирались случайным образом. Число сломанных линков изменялось в диапазоне $5, 10, 15, ..., 195$. Размеры искомых систем (параметр m) выбирались $\frac{1}{6}N_{nodes}, \frac{2}{6}N_{nodes}, ..., \frac{5}{6}N_{nodes}$.

На рисунке 4 представлено значение отношения $(c - c_{min})/(c_{max} - c_{min})$ (в процентах), характеризующее достижение какой-то характеристикой ее минимального значения, где c, c_{min} и c_{max} — выбранное, минимально и максимально полученное значение алгоритмом равномерного расширения одной из рассматриваемых характеристик c (диаметр, количество транзитных узлов и средняя загруженность линков). По графикам можно видеть, что выбранный результат минимизирует рассматриваемые характеристики во многих случаях. Приближение к максимальным значениям в некоторых случаях объясняется тем, при отборе решений происходит сортировка сначала по диаметру, затем по значениям средней загруженности линков и затем по количеству транзитных узлов. Таким образом, выбрав все системы с минимальным диаметром, возможно получить не минимальное число транзитных узлов. Для искомой системы из 228 узлов все три характеристики хорошо минимизируют-



(a) Размер искомой системы 57

(b) Размер искомой системы 114

(c) Размер искомой системы 171

(d) Размер искомой системы 228

Рис. 4. Оценка приближения значений характеристик (диаметра системы, средней загруженности линка и количества транзитных узлов) к наилучшим известным значениям в зависимости от числа сломанных линков для сети с топологией 7x7x7 и разных размеров искомой системы.

ся из-за небольшого общего количества найденных решений, что связано с отсутствием связности искомой системы при случайных отказах.

Время выполнения алгоритма на системе размера 5x5x5 составило 0.01 с — 0.1 с в зависимости от числа сломанных линков и размера искомой системы. Данное время приемлемо при выборе свободных узлов во время запуска задачи. На системах размера 7x7x7 и 9x9x9 время выполнения довольно велико: 0.7 с — 3 с и 7 с — 30 с соответственно. Планируется несколько путей решения этой проблемы. Во-первых, алгоритм можно оптимизировать в нескольких направлениях, например, нет необходимости искать все системы в малозагруженной сети; также возможно применение различных эвристик для сокращения перебора. Во-вторых, реализацию алгоритма можно оптимизировать и распараллелить с использованием технологии OpenMP.

Заключение

В данной работе описан полиномиальный алгоритм построения маршрутизируемого подмножества узлов заданного размера в сети с отказами и проведено предварительное исследование.

Алгоритм показал результаты, близкие к результатам алгоритма поиска маршрутизируемых систем в сети без занятых или отказавших узлов. Предварительное исследование результатов работы алгоритма показало, что требуется доработка алгоритма для улучшения качества результатов и увеличения скорости его работы.

В будущих работах планируется оптимизировать алгоритм и выполнить более подробное исследование.

Литература

1. Корж А.А., Макагон Д.В., Жабин И.А., Сыромятников Е.Л. Отечественная коммуникационная сеть 3D-тор с поддержкой глобально адресуемой памяти для

суперкомпьютеров транспетафлопсного уровня производительности // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной конференции (Уфа, 29 марта 2 апреля 2010 г.). Челябинск: Издательский центр ЮУрГУ, 2010. С. 227–237.

2. Жабин И., Макагон Д., Симонов А. и др. Кристалл для Ангары // Суперкомпьютеры. — 2013. — Т. зима-2013. — С. 46–49.

3. Пожилов И.А., Семенов А.С., Макагон Д.В. Алгоритм определения связности сети с топологией "многомерный тор"с отказами для детерминированной маршрутизации // Программная инженерия. — 2015. — № 3. — С. 13–19.

4. Puente V., Beivide R., Gregorio J.A., Prellezo J.M., Duato J., Izu C. "Adaptive bubble router: a design to improve performance in torus networks,"// Parallel Processing, 1999. Proceedings. 1999 International Conference on , vol., no., pp.58,67, 1999.

5. Adiga N.R., Blumrich M., Chen D. et al. Blue Gene/L torus interconnection network // IBM Journal of Research and Development. 2005. — Vol. 49, no. 2.3. — P. 265–276.

6. Scott S.L., et al. The Cray T3E Network: Adaptive Routing in a High // Performance 3D Torus. — 1996.

Approximate algorithm for choosing the best subset of nodes in the «Angara» interconnect with failures

A.V. Mukosey, A.S. Semenov, D.V. Makagon

JSC «NICEVT» (Moscow)

JSC NICEVT develops the Angara high-speed interconnect with multi-dimensional torus topology. In actual use of the "Angara"interconnect in the conditions of employment and the availability of the failed nodes arises the problem of finding an optimal nodes subset to cover a given number of nodes thus all network traffic is lying within the subset of nodes. This paper presents an approximation algorithm for solving this problem.

Keywords: Fault tolerance, communication networks, multidimensional torus, connectivity, deterministic routing, direction ordered routing.

References

1. Korzh A.A., Makagon D.V., Zhabin I.A., Syromyatnikov E.L. Otechestvennaya kommunikatsionnaya set' 3D-tor s podderzhkoy global'no adresuyemoy pamyati dlya superkomp'yuterov transpetaflopsnogo urovnya proizvoditel'nosti [Russian 3D-torus Interconnect with Support of Global Address Space Memory]. Parallelnye vychislitelnye tekhnologii (PaVT'2010): Trudy mezhdunarodnoj nauchnoj konferentsii (Ufa, 29 marta – 2 aprelya 2010) [Parallel Computational Technologies (PCT'2010): Proceedings of the International Scientific Conference (Ufa, Russia, March, 29 – April, 2, 2010)]. Chelyabinsk, Publishing of the South Ural State University, 2010. P. 527–237.

2. Zhabin, I.A. Kristall dlya Angary [Angara Chip] / I.A. Zhabin, D.V. Makagon, A.S. Simonov // Superkomp'yutery [Supercomputers]. —Winter-2013. — P. 46–49.

3. Pozhilov I.A., Semenov A.S., Makagon D.V. Algoritm opredeleniya svyaznosti seti s topologiyey "mnogomernyy tor"s otkazami dlya determinirovannoy marshrutizatsii [Connectivity problem solution for direction ordered deterministic routing in nD torus]. // Software Engineering. — 2015. — № 3. — C. 13–19.

4. Puente V., Beivide R., Gregorio J.A., Prellezo J.M., Duato J., Izu C. "Adaptive bubble router: a design to improve performance in torus networks,"// Parallel Processing, 1999. Proceedings. 1999 International Conference on , vol., no., pp.58,67, 1999.

5. Adiga N.R., Blumrich M., Chen D. et al. Blue Gene/L torus interconnection network // IBM Journal of Research and Development. 2005. — Vol. 49, no. 2.3. — P. 265–276.

6. Scott S.L., et al. The Cray T3E Network: Adaptive Routing in a High // Performance 3D Torus. — 1996.