Санкт-Петербургский Государственный Университет

Математико-механический факультет

Кафедра системного программирования

Система хранения данных. Поддержка избыточного кодирования. Поиск, сравнение и анализ применимости существующих подходов для поддержки избыточного кодирования.

Курсовая работа студента 345 группы
Котова Юрия Александровича

Научный руководитель ……. А. Н. Косякин
ведущий разработчик, ООО "Артек"

Санкт-Петербург
2010

Содержание

Введение

Во многих организациях главным активом является информация, поэтому нужно обеспечивать надежное хранение информационных ресурсов и быстрый доступ к ним.

В настоящее время повышенный интерес вызывает распределенное хранение данных, обладающее рядом преимуществ перед традиционным прямым подключением дисковых массивов к серверам: сеть хранения данных представляет собой архитектурное решение для подключения внешних устройств хранения данных, таких как дисковые массивы к серверам таким образом, чтобы операционная система распознала подключённые ресурсы как локальные.

Основные преимущества:

- Высокая масштабируемость;
- Высокая производительность и надежность;
- Простота администрирования;
- Эффективное восстановление работоспособности после сбоя;

На данный момент существует множество решений, но на небольших предприятиях они являются редкостью из-за слишком высокой стоимости. Целью проекта является создание надежной сети хранения данных, работающей на дешевом, доступном оборудовании и обладающей всеми вышеуказанными свойствами.

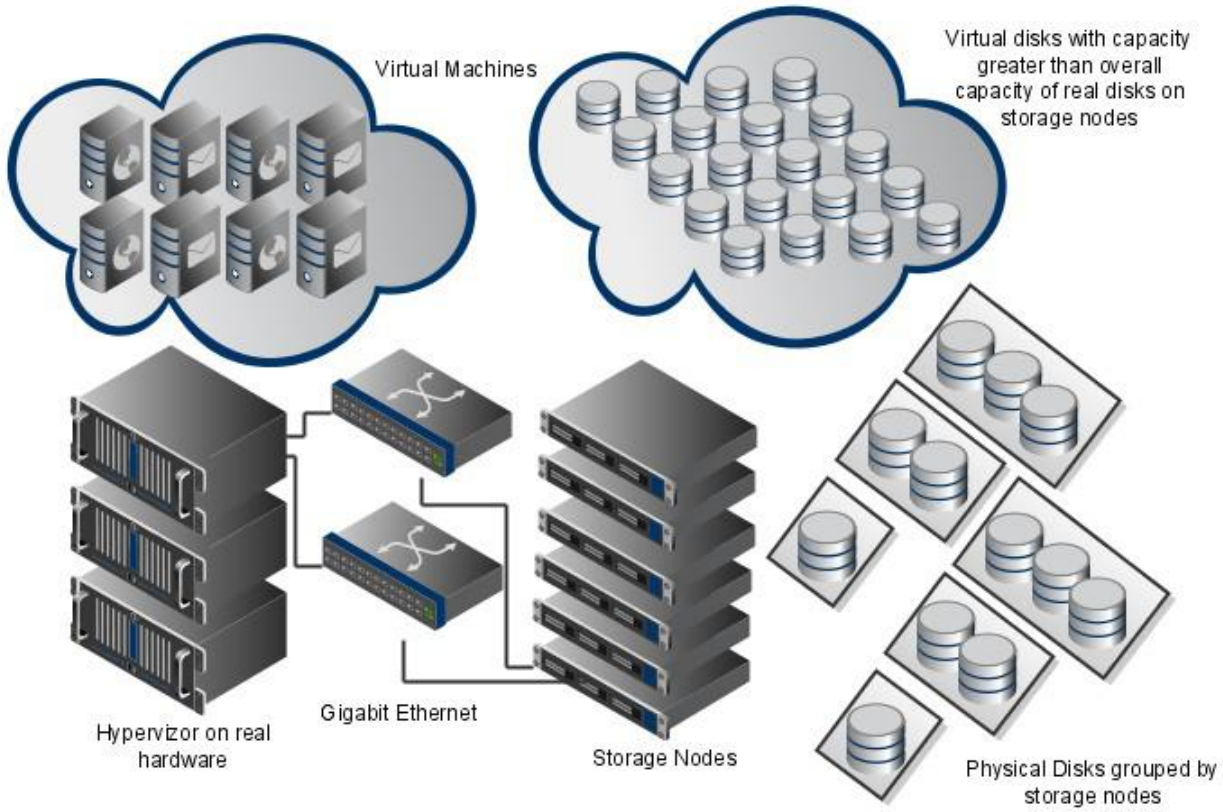

Рис.1 Распределенное хранилище

В проекте используются "Облачные вычисления", поверх некоторого числа реальных серверов, имитируется в несколько раз большее число виртуальных серверов, которые внешне ничем не отличаются от реальных. Открытым стоит вопрос, как недорого имитировать жесткие диски для этих виртуальных машин (сегодня виртуальный сервер работает на сервере 1, завтра - на сервере 2; не переставлять же диски руками!), причем при сохранении приемлемой производительности и высокой надежности, ведь современные винчестеры довольно ненадежны и могут выйти из строя, либо быть бракованными. Поэтому предлагается использовать избыточное кодирование, чтобы защитить данные от отказов узлов хранения, но при этом эффективно использовать имеющееся дисковое пространство и обеспечивать приемлемую нагрузку на CPU.

Постановка задачи

Целью данной работы является анализ подходов и алгоритмов хранения данных, сравнение различных политик хранения и анализ их применимости в проекте «Cirrostratus».

В рамках данной курсовой необходимо выполнить следующие задачи:

1. ***Выбор политик хранения***

2. ***Поиск существующих алгоритмов для каждой политики***

3. ***Анализ полученных результатов***

Выбор политик хранения.

Записывать данные на один диск без избыточности довольно опасно, поскольку диски часто выходят из строя. Поэтому нужно иметь возможность восстановить данные после выхода из строя одного или нескольких физических дисков. По этому параметру мы и различаем политики хранения данных.

Сначала определим политику хранения, используемую по умолчанию. Назовём её **«обычной».** Она будет использована в подавляющем количестве случаев, за исключением данных на виртуальных дисках, помеченных знаком «важные», поэтому она должна обеспечивать приемлемую нагрузку на CPU и достаточную надежность данных. Поскольку жесткие диски чаще всего поставляются одной партией с одними и теме же параметрами, поломка одного диска означает возможный выход из строя и ещё одного, а то и нескольких. Поэтому необходимо обеспечить жизнеспособность системы при потере хотя бы двух дисков. Алгоритмы, которые обеспечивают жизнеспособность системы при потере более двух дисков, дают весьма ощутимую нагрузку на CPU и вероятность потери более двух дисков до того как система успеет восстановить утраченную информацию (ориентировочно не более суток) довольно мала.

Итак, выбор для **«обычной»** политики хранения – алгоритмы, которые обеспечивают жизнеспособность системы при потере двух дисков (**RAID-6**).

Далее рассмотрим политику хранения важной информации, точнее политику хранения, применяемую к виртуальным дискам, помеченных знаком «важные». Назовём её **«важной»**. Она будет применяться не так часто, поэтому нагрузку на CPU можно не брать во внимание.

Для такой политики хранения информации может применяться технология зеркалирования, что позволит сэкономить на нагрузке на CPU, но займет много дискового пространства, либо можно использовать алгоритмы, которые обеспечивают жизнеспособность системы при потере более двух дисков, которые, наоборот, экономно используют дисковое пространство, но нагружают CPU. Мы выбираем второй вариант, так как при зеркалировании данных даже на четыре диска, эти диски распределяются алгоритмом CRASH и могут оказаться на одном узле хранения, который тоже может выйти из строя, что приведёт к потере данных.

Поиск существующих алгоритмов.

Алгоритмы для «обычной» политики

Данные алгоритмы, с возможностью восстановления после потери двух дисков, так же называются RAID-6 кодами.

Среди представителей специфичных RAID-6 кодов стоит отметить трёх: EVENODD [1], Row Diagonal Parity (RDP) [4] и Minimal Density [2, 5, 6] RAID-6 коды.

Алгоритмы для «важной» политики

Алгоритмы, обеспечивающие жизнеспособность системы при потере достаточно большого числа дисков, это: Classic Reed-Solomon коды [7] и Cauchy Reed-Solomon коды [3].

Анализ работы алгоритмов

Наша система хранения данных состоит из массива из n дисков одинакового размера. Из этих n дисков k используются для хранения данных, а остальные m содержат избыточную информацию, их так же называют parity, которые рассчитываются по данным. Мы пометили диски с данными $D_0, \ldots, D_{k-1}$ и parity диски $C_0, \ldots, C_{m-1}$ соответственно. Типичная система изображена на рисунке 2.

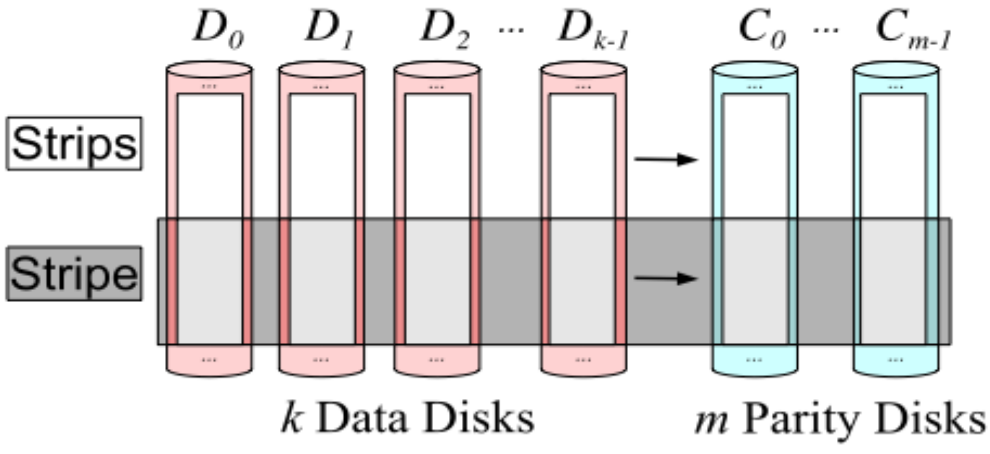

Рис.2Общий принцип избыточного кодирования.

Каждый диск разделён на блоки фиксированного размера. Каждый parity блок (strip) кодируется, используя один блок с каждого диска данных, и такая коллекция k + m блоков называется полосой (stripe). Каждая полоса кодируется и декодируется независимо от остальных.

Reed-Solomon (RS) коды

В данных кодах блоки состоят из w-битных слов, где w должно быть достаточно большим, чтобы $n \leq 2^w + 1$. W обычно выбирают кратным длине машинного слова: $w \in \{8, 16, 32, 64\}$. Однако его можно выбирать произвольно. Reed-Solomon коды рассматривают каждое слово как число в диапазоне от 0 до $2^w - 1$ и оперируют с ними как числами из поля Галуа ($GF(2^w)$), которое определяет сложение, умножение и деление с этими словами так, что система получается замкнутой. Кодирование Reed-Solomon кодом строится на фактах линейной алгебры. Кодирующая матрица (G^T) строится по матрице Вандермонда, и эта матрица умножается на k слов данных, и получается k + m слов, данных и избыточных соответственно. При сбое дисков декодирование происходит путём удаления строк из матрицы G^T, инвертирования её и умножения на инвертированные уцелевшие слова. Этот процесс эквивалентен решению системы независимых линейных уравнений. Построение G^T по матрице Вандермонда гарантирует, что инвертированную матрицу всегда можно вычислить.

Cauchy Reed-Solomon (CRS) коды

CRS коды это модификация RS кодов. Во-первых, для построения G^T матрицы используется матрица Коши, а не Вандермонда. Во-вторых, заменены дорогие умножения, использовавшиеся в RS кодах, на специальные XOR операции. Такая модификация заменяет G^T с $n \times k$ матрицы, состоящей из w-битных слов, на $wn \times wk$ матрицу, состоящую из битов. Также как и для RS кодов,w нужно выбирать, чтобы выполнялось условие $n \leq 2^w + 1$.

Вместо действий с отдельными словами CRS коды оперируют целыми блоками. В частности, блоки разделены на w пакетов, и эти пакеты могут быть достаточно большими. Теперь процесс кодирования включает только XOR операции, избыточный пакет получается операцией XOR от всех пакетов данных, которые соответствуют единицам в соответствующей строке матрицы G^T. Этот процесс показан на рисунке 3, который иллюстрирует как вычисляется последний избыточный пакет.

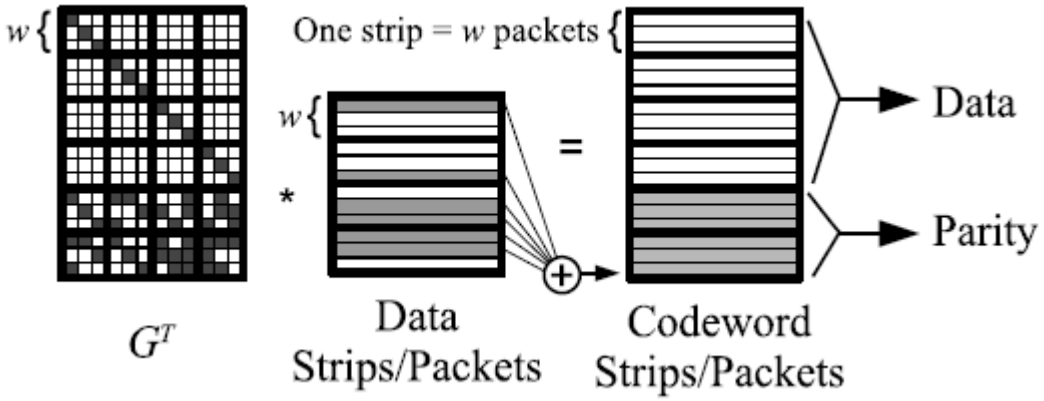

Рис.3 Пример CRS кода для k = 4 и m = 2.

Для эффективного выполнения операций XOR длина пакета должна равняться длине нескольких машинных слов. Размер блока равен w*(размер пакета). Теперь w не относится к машинному слову и не ограничено [8, 16, 32, 64]; наоборот, любое значение может быть выбрано до тех пор, пока $n \leq 2^w$.

Декодирование в CRS кодах аналогично RS кодам, строки соответствующие испорченным данным, удаляются из матрицы G^T, матрица транспонируется и используется для восстановления потерянных данных. Получается, что производительность кода напрямую связана с количеством единиц в матрице G^T.

EVENODD и RDP

Оба кода, EVENODD и RDP, являются типичными представителями RAID-6 кодов. Условно в RAID-6 первый избыточный диск обозначается P, а второй Q. EVENODD и RDP основываются на таком же принципе, что и CRS коды, блоки составляются из w пакетов. В EVENODD w выбирается так, чтобы:

1) $k + 1 \leq w$

2) w+1 являлось простым числом.

В RDP w+1 должен быть простым числом, и $k \leq w$. Производительность обоих кодов лучше всего, когда $(w-k)$ минимально.

Minimal Density RAID-6 коды

Для RAID-6 систем матрица G^T довольно ограничена. В частности, первые kw строки матрицы G^T составляют единичную матрицу, а в таком случае следующие w строк должны содержать k единичных матриц. Только последние w строк можно варьировать. В работе [2],

Blaum и Roth продемонстрировали, что при $k \leq w$ эти оставшиеся w строки должны содержать по крайней мере $kw + k - 1$ единиц. Коды, содержащие ровно столько единиц, называются Minimal Density (коды наименьшей плотности).

Заключение.

В данной работе был проведён анализ применимости алгоритмов избыточного кодирования в проекте «Cirrostratus». Также были рассмотрены различные политики хранения данных.

В курсовой работе был произведен анализ работы алгоритмов. Выявлены их параметры и ограничения. Обозначены возможные пути оптимизации рассматриваемых алгоритмов. Полученные результаты приведены в таблице 1.

Название кода	**Кол-во вост. дисков**	**Параметры**	**Ограничения**
Reed-Solomon	m	- w – число бит в слове	- $n \leq 2^w+1$
Cauchy Reed-Solomon	m	- w – число пакетов - размер пакета	- $n \leq 2^w+1$
EVENODD	2	- w – число пакетов - размер пакета	- $k+1 \leq w$ - w+1 - простое
Row Diagonal Parity (RDP)	2	- w – число пакетов - размер пакета	- $k \leq w$ - w+1 - простое
Minimal Density RAID-6	2	- w – число пакетов - размер пакета	- последние w строк матрицы G^T содержат ровно kw+k-1 единиц

Табл.1 Сравнительный анализ алгоритмов.

Список литературы

[1] BLAUM, M., BRADY, J., BRUCK, J., AND MENON, J. EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures. //IEEE Transactions on Computing 44, 2 (February 1995), 192– 202.

[2] BLAUM, M., AND ROTH, R. M. On lowest density MDS codes. //IEEE Transactions on Information Theory 45, 1 (January 1999), 46–59.

[3] BLOMER, J., KALFANE, M., KARPINSKI, M., KARP, R., LUBY, M., AND ZUCKERMAN, D. An XOR-based erasure-resilient coding scheme. //Tech. Rep. TR-95-048, International Computer Science Institute, August 1995.

[4] CORBETT, P., ENGLISH, B., GOEL, A., GRCANAC, T., KLEIMAN, S., LEONG, J., AND SANKAR, S. Row diagonal parity for double disk failure correction. //3rd Usenix Conference on File and Storage Technologies (San Francisco, CA, March 2004).

[5] PLANK, J. S. A new minimum density RAID-6 code with a word size of eight. //NCA-08: 7th IEEE International Symposium on Network Computing Applications (Cambridge, MA, July 2008).

[6] PLANK, J. S. The RAID-6 Liberation codes. //FAST-2008: 6th Usenix Conference on File and Storage Technologies (San Jose, February 2008), pp. 97–110.

[7] REED, I. S., AND SOLOMON, G. Polynomial codes over certain finite fields. //Journal of the Society for Industrial and Applied Mathematics 8 (1960), 300–304.