Компьютерная Математика

УДК 681.3.06

В.В. ФАЛЬФУШИНСКИЙ

РЕАЛИЗАЦИЯ МОДЕЛЕЙ ХРАНИЛИЩА В СЕМЕЙСТВЕ КЛАСТЕРНЫХ КОМПЛЕКСОВ ИНПАРКОМ

Рассмотрены архитектуры хранилища для кластерных комплексов. Приведена сравнительная характеристика хранилища для кластерных комплексов.

Введение. Институт кибернетики имени В.М. Глушкова НАН Украины вместе с НВО «Електронмаш» создал первое в Украине семейство серийных кластерных комплексов Инпарком-16, -64, -128 и -256. На этих кластерных комплексах реализовано ряд библиотек для решения важных классов задач. Сокращение времени постановки задачи, исследование и решение задач достигается не только за счет организации параллельных вычислений, а также за счет интеллектуального программирования, которое учитывает адаптивную настройку алгоритма и синтезированной программы со свойствами задачи с учетом технических и математических возможностей кластера [1, 2].

С развитием дешевых систем коммуникации, высокопродуктивных технологий и высокоскоростных сетей одним из ключевых моментов в оптимизации процесса решения задачи занимает быстрое считывание и записывание данных. Таким образом, появляется задача подбора подходящей архитектуры хранилища, которое учитывает оптимальный выбор архитектуры файловой системы операционной среды и технологий реализации.

Дальше проанализируем разные аспекты выбора оптимальной архитектуры хранилища для кластерного комплекса выходя с потребностей к реализованным в составе ОС штатных файловых систем, доступных на ИТ рынках Украины.

Мобильность открытых систем. С развитием открытых систем стало возможным распространение архитектуры «клиент-сервер». Основным смыслом подхода открытых систем является упрощение комплексирования

вычислительных систем за счет международной и национальной стандартизации аппаратных и программных интерфейсов. Основной причиной развития концепции открытых систем явились повсеместный переход к использованию локальных компьютерных сетей и проблемы комплексирования аппаратно-программных средств. В связи с бурным развитием технологий глобальных коммуникаций открытые системы приобретают еще большее значение и масштабность.

Технологии и стандарты открытых систем обеспечивают реальную и проверенную практикой возможность производства системных и прикладных программных средств со свойствами мобильности (portability) и интероперабельности (interoperability). Свойство мобильности означает сравнительную простоту переноса программной системы в широком спектре аппаратно-программных средств, соответствующих стандартам. Интероперабельность означает упрощение комплексирования новых программных систем на основе использования готовых компонентов со стандартными интерфейсами.

Использование подхода открытых систем выгодно и производителям, и пользователям. Открытые системы обеспечивают естественное решение проблемы поколений аппаратных и программных средств. Производители таких средств не вынуждены решать все проблемы заново; по крайней мере, они могут временно продолжать комплексировать системы, используя существующие компоненты [3].

Особенности файловых систем ОС. Единицей обмена в компьютерной сети является файл. Его спецификацию понимают все ОС, однако каждая ОС имеет специфику восприятия файлов. Большие объёмы файлов могут быть сохранены в хранилищах. Таким образом, хранилища исполняют функцию размещения информации в блоках определенного объема, а файловая система выполняет функцию промежуточного шара между пользователем и хранилищем, вводит абстракцию файлов и каталогов для пользователя, обеспечивает управление доступом к файлам и устойчивой системой восстановления утерянных или нарушенных файлов во время сбоев или потери питания.

В ОС файлы имеют два стандартные атрибута: идентификатор пользователя (user-id) и группы (group-id). Такие атрибуты допускают легко разграничивать доступ на считывания, записывания и выполнения для отдельных пользователей и групп. Также возможно управлять доступом для всех и определенных групп. Поэтому, как только процесс делает запрос на доступ к файлу, файловая система проверяет его на возможность доступа, а уже потом возвращает дескриптор файла. Когда некоторое количество процессов получают доступ к файлу, то каждый процесс получает свой дескриптор для роботы с файлом.

Файловая система поддерживает разные типы файлов: регулярные файлы, которые могут выполнять определенный набор функций и хранимые файлы, которые размещаются в памяти (т.е. для считывания и записывания). Другими типами файлов есть сокет-файлы (socket files) и файлы устройств. Сокет-файлы содержат информацию о межпроцессном взаимодействии разных приложений, а файлы устройств – о взаимодействии между сущностями пользовательского уровня (user-level entities) и ядра (kernel-level entities).

Большинство файловых систем встроено в ядро ОС. Интерфейсом для доступа к файлам является виртуальная файловая система (VFS), задекларированная в стандарте POSIX [4], куда входят:

- основные определения, список основных определений и соглашений, используемых в спецификациях заголовочных файлов языка Си, которые предоставляются в соответствии со стандартом;
- оболочка и утилиты, описание утилит и командной оболочки sh, стандарты регулярных выражений;
- системные интерфейсы, список системных вызовов языка Си;
- обоснование, объяснение принципов, используемых в стандарте.

Поскольку файловая система относится к уровню прикладного программного обеспечения, реализуя системные интерфейсы и обеспечивая мобильность данных, она может работать в среде разных операционных систем. Системные интерфейсы предлагают наборы сигнатур функций, построенных в соответствии со стандартом POSIX [4].

Поскольку файловая система сохраняет файлы на диске блоками, то скорость доступа к этим блокам имеет существенное значение для считывания, записывания и поиска данных. На физическом уровне такой показатель зависит от скорости прокручивания диска. Но такие алгоритмы обратной записи (write-behind) и обратного считывания (read-behind), которые относятся к уровню файловой системы, также влияют на скорость работы с данными. Технология предупреждающего записывания обеспечивает запоминание операций процесса и выполнения их независимо от процесса записывания или считывания [5].

Современные архитектуры организации хранилища. С ростом количества узлов кластера и объемов данных возрастает проблема выбора архитектуры хранилища. Хранилище кластера представляет собой отдельную систему, которая предоставляет свои ресурсы для хранения данных. Известны такие архитектуры хранилищ:

- хранилище с прямым доступом (DAS – Direct attached Storage);
- хранилище из сетевым доступом (NAS – Network attached Storage);
- кластерный NAS (CNAS – Clustered NAS);
- объектное хранилище (OSS – Object Storage Systems);
- сетевая файловая система (NFS – Network File System);
- параллельная файловая система (PFS – Parallel File System).

Наиболее распространенные архитектуры – DAS, NAS, NFS и PFS. Как показали предварительные исследования, одна архитектура не может быть универсальной для всех задач, поэтому разработчики кластерных комплексов комбинируют несколько архитектур вместе.

В процессе проектирования и настройки прикладной программы для вычислений на кластере целесообразно рассмотреть разные методы доступа к данным. Для этого наиболее подходят некоторые из возможных архитектур организации хранилища, скорость доступа к данным, стоимость эксплуатации и рыночная цена хранилища, его объем и механизмы поддержки целостности данных [6, 7]. Рассмотрим три потенциальных архитектуры хранилища кластера ИНПАРКОМ: DAS, NAS+NFS и PFS.

Хранилище с прямым доступом (DAS – Direct attached Storage). Такая архитектура хранилища представляет собой модуль ОС, который состоит с пользовательского модуля ОС и модуля хранилища. Пользовательский модуль обеспечивает контроль доступа, возможности записывания и считывания и управления квотами. Модуль хранилища транслирует запросы с пользовательского модуля в секторы, блоки и цилиндры, которые интерпретируют функции устройства хранения информации. Хранилище с прямым доступом можно подсоединить к узлу с помощью канала SCSI или оптоволоконного соединения. Основными файловыми системами, которые поддерживают работу с такой архитектурой хранилища, – ext3, reiserFS, NTFS (для Windows), SGI XFS. Такая архитектура хранилища имеет показатели надежности, помехоустойчивости, относительно высокой скорости доступа к данным, ограничены пропускной способностью канала передачи данных. Основные недостатки такого хранилища – цена и отсутствие мобильности, а также существенное возрастание нагрузки сети в случае синхронизации данных между узлами.

NAS с поддержкой NFS. Хранилище такой архитектуры является самым распространенным, если учесть его относительно простую и дешевую архитектуру. Оно состоит из отдельного сервера, клиента и сети, которая их соединяет. Таким образом, клиент делает запрос файловой системы через сеть Gigabit Ethernet или InfiniBand, сервер обрабатывает запрос и передает результат клиенту. Так сервер наследует большинство характеристик хранилища с прямым доступом. Такая архитектура помогает избежать проблемы дублирования файлов в хранилище, переносом операций с файлами на сторону сервера, применения кэширования для часто повторяющихся запросов. Хотя есть ряд проблем – ограничение доступа к хранилищу, ограничения доступа из-за пропускной способности сети. Самые распространенные файловые системы такой архитектуры – NFS и CIFS.

Параллельная файловая система (PFS – Parallel File System). Когда стоит задача построения большого и в то же время быстрого хранилища с минимальными затратами, то большинство выбирает параллельную файловую систему. Согласно статистическим данным top500 за август 2008 г., 6 из 10 кластеров используют параллельную файловую систему [6], которая позволяет соединить определенное количество узлов кластера в массив и таким образом получить хранилище с довольно большим объемом и высокой скоростью доступа.

Параллельная файловая система состоит из клиента и сервера. Сервер предоставляет пространство для сохранения данных, а клиент использует его для сохранения данных. Узлы кластера могут быть как клиентами, так и серверами, если их часть собственного дискового пространства добавляется к пространству параллельной файловой системы.

Для сохранения служебной информации элементов файловой системы (имена файлов, права доступа, атрибуты) используют сервер метаданных. На уровне ОС параллельная файловая система виртуальная, потому она не привязана к типам физических разделов на диске. Для функционирования такой системы используют выделенную администратором область существующей файловой

системы (каталог, файл), например "/tmp/storage", доступ к которой получают через специальный модуль ядра системы и демон виртуальной файловой системы. Для реализации параллельной файловой системы используют Filesystem in Userspace (FUSE).

Параллельные файловые системы используют разные алгоритмы сортировки данных в хранилище. Для ускорения доступа необходимо равномерно распределять данные между дисками / узлами кластера для лучшего параллельного выполнения файловых операций. Но оптимизация разделения данных с позиций оптимизации доступа – довольно сложная задача. Потому чаще всего используют последовательное сохранение файлов в хранилище.

Известен ряд параллельных файловых систем, разработанный для разных типов кластерных комплексов: GlusterFS, SSHFS, GmailFS, EncFS, NTFS-3G, WikipediaFS, Lustre. Они позволяют строить свои архитектуры хранилищ, с разным доступом к данным в процессе их обработки. Некоторые файловые системы представляют свой набор функций для управления процессом параллельного доступа к данным во время выполнения программы. Скорость доступа в параллельной файловой системе возрастает с ростом количества узлов. Единственная проблема параллельной файловой системы – восстановление системы после сбоев.

Результат задачи выбора. Для выбора оптимальной архитектуры хранилища было реализовано три архитектуры (таблица). Для хранилища с прямым доступом в Инпарком-256 (как наиболее мощному вычислительному комплексу семейства) использованы жесткие диски размером 250 Gb с файловой системой ext3. Для реализации NAS+NFS использовано хранилище объемом 931 Gb, связано с узлами через сети Infiniband. Для реализации хранилища с параллельной файловой системой было использовано 8 узлов с жесткими дисками по 250 Gb, соединенными с помощью сети Infiniband, которая имеет пропускную способность 20 Гбит/с.

ТАБЛИЦА. Характеристики файловых систем

Архитектура	Скорость доступа к данным, Мб/с	Цена, $	Объем хранилища	Уровень целостности данных
Хранилище с прямым доступом	74,29	60 / 1 узел	250 Gb	Максимальный
NAS с поддержкой NFS	68,11	800	1Tb	Средний
Параллельная файловая система	71,27	0	4Tb	Минимальный

Исходя из данных таблицы невозможно сделать однозначный выбор файловой системы. Получив выигрыш в некоторых показателях, таких как скорость и цена, можно получить существенные потери при сбоях системы и наоборот. Поэтому оптимальным решением будет использование определенных систем в зависимости от параметров задачи.

Выводы. С проблемой эффективности обработки исходных данных разработчики сталкиваются каждый раз. Изучение этой проблемы способствовало сужению круга архитектур файловых систем и нахождения оптимальных, в зависимости от определенных критериев. Изложенные в работе архитектуры были протестированы на кластере Инпарком-256 и могут быть укомплектованы к любому другому кластеру.

В.В. Фальфушинський

РЕАЛІЗАЦІЯ МОДЕЛЕЙ СХОВИЩА У СІМЕЙСТВІ КЛАСТЕРНИХ КОМПЛЕКСІВ ІНПАРКОМ

Розглянуто можливість застосування різних моделей сховища. Можливість оптимізації доступу до данних за допомогою застосування різних архітектур сховища, які відповідають вимогам малої ціни, великого об'єму, високої стабільності та високої швидкості доступу. Продемонстровано швидкість роботи різних архітектур сховища.

V.V. Falfushinsky

IMPLEMENTATION OF STORAGE MODELS IN THE FAMILY OF INPARCOM CLUSTERS

A possibility of using different models of file systems is described. Optimization of data access with the help of different storage architectures with criteria of small price, big size, stability and access speed is analyzed.

1. *Численное* программное обеспечение интеллектуального MIMD-компьютера ИНПАРКОМ / А.Н. Химич, И.Н. Молчанов, В.И. Мова и др.– Киев: Наук. думка, 2007. – 222 с.
2. *Перевозчикова О.Л., Тульчинский В.Г., Ющенко Р.А.* Построение и оптимизация параллельных компьютеров для обработки больших объемов данных // Кибернетика и системный анализ. – 2006. – № 4. – С. 117–129.
3. *Noronha R* .Designing high-performance and scalable file network attached storage with iInfiniband // Technical Report OSU-CISRC-3/08-TR09,Oniversity, 2008. – P. 244.
4. *ДСТУ* 4249:2003. *Настанова* щодо POSIX-сумісних середовищ відкритих систем (POSIX-OSE) / О. Демська-Кульчицька, А. Гречко, Б. Кульчицька, О. Перевозчикова, В. Січкаренко. – К.: Держспоживстандарт, 2007. – 176 с.
5. *Тульчинский В.Г., Чарута А.К.* Оценка времени обработки данных в кластерных системах // Проблеми програмування. – 2006. – № 2–3. – С. 118 – 123.
6. www.top500.org , TOP 500 of supercomputers, 2008.
7. *Липаев В.В., Филинов Е.Н.* Мобильность программ и данных в открытых информационных системах. – М.: Научная книга, 1997. – 368 с.

Получено 02.11.2008

Об авторе:

Фальфушинский Владислав Владимирович,
инженер-программист 1-й категории
Института кибернетики имени В.М. Глушкова НАН Украины.