Операционная система для К компьютера

•Jun Moroo •Masahiko Yamada •Takeharu Kato
FUJITSU Sci. Tech. J., Vol. 48, No. 3, pp. 295–301 (July 2012)

Для достижения К компьютером наивысшей в мире производительности Fujitsu работал над следующими тремя улучшениями при разработке операционной системы (OS). Во первых, для выявления максимальной аппаратной производительности наших оригинальных CPU и интерконнекта, мы разработали механизм управления аппаратными расширениями непосредственно из приложений. В качестве второго улучшения мы предложили функцию планирования синхронизаций, которая минимизирует время ожидания для синхронизации параллельно исполняемых програм, получающийся из прерываний системы координацией времени выполнения заданий и системы между многими узлами. Третье: для улучшения производительности доступа к памяти и эффективности ее использования была достигнута поддержка множественного размера страниц памяти, которая позволяет использовать более одного размера страниц. Данная статья также описывает функции повышения производительности, удобства в использовании и надежности при разработке ОС.

1. Введение

Суперкомпьютеры используются крупномасштабных вычислительных моделированиях в различных областях науки и техники. Для предсказания погоды, например, область исследования может быть разделена на муниципальном уровне или даже глубже до уровня станций наблюдения для вычисления таких значений как температура, скорость ветра для каждой подобласти, что позволяет делать очень точные прогнозы. Аэродинамическое моделирование в отношении воздушных судов позволяет персоналу оценить как самолет будет вести себя в полете без построения модели или реального летательного аппарата. При разработке новых лекарственных средств, вещества являющиеся претендентами на медицинские цели могут быть извлечены из огромного числа типов и комбинаций для их сокращения к представителям с лечебным эффектом.

Проведение таких масштабных моделирований вовлекает в процесс огромные объемы памяти и вычислительных узлов, а также огромное количество файлов данных. Соответственно, существенной является способность операционной системы (ОС) доставлять до приложений максимум производительности аппаратных средств.

Данная статья приводит описание ОС К компьютера[note)], который обладает способностью проводить крупномасштабные моделирования.

2. Конфигурация программного обеспечения К компьютера

Обычно суперкомпьютер выполняет высокопроизводительные вычисления посредством:

1) Разделения огромных вычислений
2) Одновременной скоординированной работы большого количества компьютеров (вычислительных узлов)
3) Передачи результатов вычислений на высоких скоростях между вычислительными узлами с использованием набора стандартных библиотек обмена данными под названием MPI (Message Passing Interface).

ОС осуществляет запуск приложений и обработку запросов ввода/вывода, а также руководит системными процессами, включающими контроль за временем дня. Для ОС К компьютера мы максимизировали производительность аппаратных средств и программного обеспечения (ПО) путем улучшения ядра ОС и библиотек. Мы работали над развитием ОС К компьютера, имея целью достижения 10PFLOPS, уровень производительности которого в десять раз выше, чем производительность системы в момент первоначальной разработки.

[note)] “K computer” - английское название, которое RIKEN использовал для суперкомпьютера в данном проекте начиная с июля 2010. “K” пришло из японского слова “Kei,” которое обозначает 10пета или 10 в 16й степени.

Основное программное обеспечение К компьютера состоит из ОС и базового промежуточного ПО (ПО управления работ, поддержка языков программирования) (Рисунок 1). ОС реализует часть ядра Linux, зависящую от архитектуры и дополнительные драйверы для использования аппаратных средств К компьютера, так что они могут быть использованы начиная с промежуточного ПО в ПО более высоких уровней.

Мы работали над разработкой ОС для установки на К компьютере имея целями: улучшенное использование, улучшенную производительность и улучшенную надежность, как показано в Таблице 1. В Следующих разделах представлены соответствующие цели.

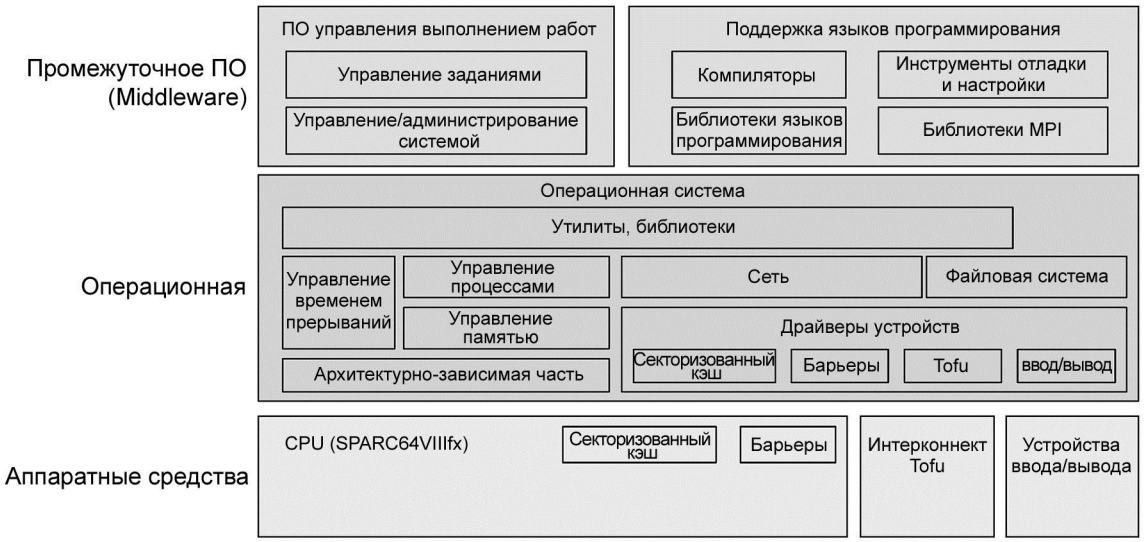

Рисунок 1
Конфигурация программного обеспечения К компьютера

Таблица 1
Свойства операционной системы К компьютера.

Элемент	Описание
Кол-во одновременных обменов данными	4 соединения на передачу + 4 соединения на прием
Рабочая частота	312.5MHz
Пропускная способность коммутации	100GB/s
Скорость соединения	5GB/s в двух направлениях
Кол-во портов	10
Технологический процесс	65-nm CMOS
Размер кристалла	18.2mm x 18.1mm
Кол-во логических вентилей	48 миллионов вентилей
Кол-во ячеек SRAM	12 миллионов бит
Различные шины ввода/вывода	
Соединение Tofu	6.25Gb/s, 80lanes
Шина процессора	6.25Gb/s, 32lanes
PCI Express	5Gb/s, 16lanes

3. Улучшенное использование (эффективное использование пользовательских средств)

Некоторые из существующих суперкомпьютеров используют специализированные ОС, на которых программы, разработанные некоторыми пользователями не могут использоваться. Для ОС К компьютера мы адаптировали Linux, обеспечивая дальнейшее использование пользовательских средств.

Аналогично обычным ОС, ОС К компьютера имеют средства управления процессами, памятью, операциями ввода/вывода (драйверы устройств), файловой системой и сетевыми функциями. Linux используется более чем в 91% ранга суперкомпьютеров TOP500 и мы остановились на ней с точки зрения возможности портации приложений пользователей. На К компьютере пользователи могут использовать POSIX API, интерфейс прикладного программирования (API) UNIX.

Кроме того, ОС К компьютера имеет собственное расширение API (библиотеки, утилиты) для использования расширений аппаратных средств, которые позволяют легкое использование аппаратных функций через компиляторы и библиотеки функций MPI. Эти ухищрения позволяют ОС К компьютера портировать общее ПО с открытым исходным кодом (прикладные программы, инструментарий и т.п.) путем перекомпиляции без необходимости внесения изменений в исходную программу.

4. Улучшение производительности

4.1 Непосредственное управление расширениями аппаратных средств

Для ускорения SPARC64 VIIIfx является CPU на основе SPARC64 VII, оснащенного расширенным набором регистров, расширенных инструкций SIMD, аппаратного барьера между ядрами и секторизованного кэша. Для ОС К компьютера мы подготовили механизм с дополнительными драйверами устройств, которые позволяют непосредственное управление из приложений этими аппаратными расширениями. Драйверы отображают регистры, которые управляют барьерной синхронизацией и секторизацией кэша в пространство памяти приложения, с помощью чего достигается высокоскоростное управление без привлечения дополнительных системных вызовов.

Секторизация кэша предназначена для разделения кэша в CPU на две виртуальные области для облегчения кэширования повторно используемых приложениями данных.

В частности, компилятор генерирует инструкции доступа к памяти для кэширования реже используемых данных в секторе 0, в то время как чаще используемые данные сохраняются в секторе 1. Это предотвращает выталкивание из кэша часто используемых данных и, таким образом, повышает производительность выполнения приложений. Количество часто используемых данных может изменяться в зависимости от функций в приложении. Соответственно, отношение секторизации кэша используемого пользовательским приложнием было сделано изменяемым во время выполнения программы, что позволяет оптимизировать эффективность использования кэша.

Нельзя достичь достаточного эффекта повышения производительности управления кэш-памятью, если при выполнении операции секторизации кэша осуществляется вызов драйвера устройства, что влечет дополнительные накладные расходы. По этой причине драйверы устройств для К компьютера были разработаны и реализованы таким образом, чтобы позволить пользовательским приложениям непосредственно управлять регистрами, которые регулируют секторизацию кэш-памяти. Это успешно привело к уменьшению времени управления секторизацией кэша с нескольких микросекунд до нескольких наносекунд.

4.2 Улучшение производительности доступа к памяти

ОС К компьютера оснащена поддержкой больших страниц памяти, как функцией, которая одновременно обеспечивает высокую производительность доступа к памяти, так и высокую эффективность ее использования.

Обычно блок управления виртуальной памятью, предоставляемый для Linux с 64-bit SPARC управляет памятью, используемой программами, страницами памяти с размером 8Kbyte.

Кэш (TLB: translation lookaside buffer, буффер ассоциативной трансляции) предназначен для ускорения преобразования адресов, выполняемое аппаратными средствами, однако количество записей ограничено. При размере страницы в 8Kbytes использование памяти в несколько Mbytes или более заставляет процесс превышать емкость TLB и в результате производить накладные расходы в результате трансляции адресов.

Если увеличить размер страницы, управляемый одной записью, то с помощью TLB может быть достигнут высокоскоростной доступ к памяти в несколько Gbytes и более. Однако, управление большими страницами всей областью данных страницами одного размера может снизить эффективность использования памяти из-за неиспользуемой на больших страницах памяти. Для решения этой проблемы ОС К компьютера позволяет опционально устанавливать размер страницы 4Mbytes, 32Mbytes и 256Mbytes

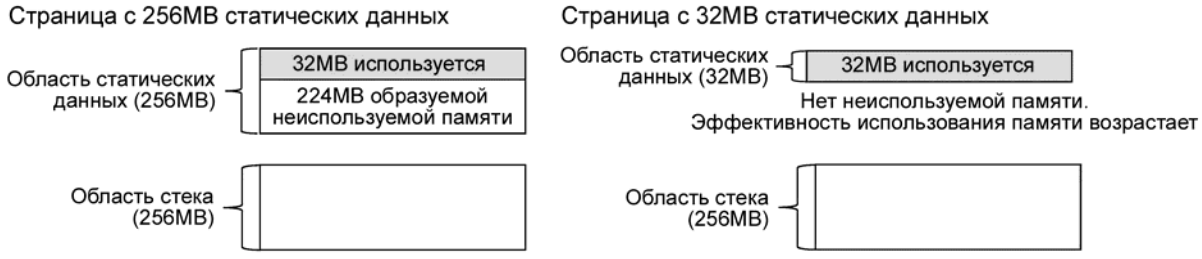

Рисунок 2
Эффект от поддержки множественного размера страниц в процессе,
который использует 32MB статических данных

Например, для запуска приложения, использующего стек больше чем область статических данных, включающих глобальные переменные, для стека может быть выделена большая страница, а маленькая страница для области данных. Таким образом может быть достигнуто одновременное улучшение и эффективности использования памяти и производительности, благодаря уменьшению числа промахов TLB (Рисунок 2).

4.3 Улучшение масштабируемости.

С целью повышения производительности параллельных заданий на множестве узлов (улучшение масштабируемости), мы разработали планировщик синхронизации и механизм отбора статистических выборок

4.3.1 Планировщик синхронизации

Отдельное вычислительное задание, запущенное на системе К компьютере, является параллельной программой, развернутой на множестве узлов и всякий раз, когда завершается определенный вычислительный процесс, имеет место взаимная синхронизация и обмен данными между узлами, соединенными с помощью интерконнекта.

Для функционирования системы необходимо запустить ОС и работу демонов ПО (программ, которые работают в фоновом режиме, обеспечивая управление системой), а они работают асинхронно, независимо от задания.

Эти операции демонов вызывают прерывания (системный шум) параллельной программы. В вычислениях на десятках тысяч узлов прерывания могут увеличиваться пропорционально числу узлов, приводя к значительному снижению производительности.

Существует два способа решения данной проблемы:

1) Выделение определенных ядер узла, состоящего из множества процессорных ядер, для запуска на них операционной системой демонов с целью устранения прерываний в задании на каждом узле (пространственное разделение)
2) Уменьшение прерываний в синхронизации/обмене данными параллельной программы до определенного уровня независимо от числа узлов при помощи синхронизации работающих демонов между вычислительными узлами (метод планировщика синхронизации) (временно́е разделение)

С использованием метода, описанного в пункте 1), когда одно из восьми ядер процессора К компьютера выделяется системе и семь ядер заданию, эффективность выполнения ограничивается 87.5% (семь восьмых). Соответственно, мы разработали для К компьютера описанный в пункте 2) метод планировщика синхронизации.

Кроме планировщика синхронизации К компьютера, ОС использует функцию барьера интерконнекта Tofu для достижения синхронизации между узлами. Планировщик синхронизации увеличивает работу функции планировщика Linux для синхронной работы на интервале в 100мс, например, и запускает демоны на 1й мс после 100мс. Параллельная программа может работать 99% времени синхронно со всеми узлами без прерываний, что позволяет улучшить производительность пропорционально числу узлов (Рисунок 3).

4.3.2 Функция статистических выборок

К компьютер сам уменьшает прерывания с использованием планировщика синхронизаций уменьшая воздействие демонов на задания и оптимизируя их работу. В качестве примера, данная секция описывает улучшение функции статистических выборок.

При выполнении пользовательского приложения на вычислительном узле периодически запускаемый в системе процесс- демон может вызывать прерывания (системный шум) в пользовательском приложении и, в конечном итоге, к изменению производительности пользовательского приложения (производительность выполнения может изменяться в зависимости от времени исполнения этого пользовательского приложения).

По этой причине уменьшение системного шума на вычислительном узле требует сведения к минимуму выполнение демонов, которые требуются периодически.

Для мониторинга с целью осуществления нормальной работы системы и анализа бутылочных горлышек производительности ОС выполняет статистические выборки, включающие использование CPU и памяти. Как правило, с использованием *cron* запускается утилита *sadc*- предназначенная для периодического выполнения системных процессов, для чтения данных ядра и записи файла статистических данных. Этот метод не вызывает значительного шума в небольших системах, однако в К компьютере, который является огромной системой, этот процесс статистических выборок может приводить к системному шуму.

Для К компьютера была разработана удаленная функция *rsadc* (remote *sadc*), которая собирает статистику производительности вычислительных узлов в узлах ввода/вывода (Рисунок 4). Мы задались целью уменьшения стоимости и увеличения надежности К компьютера путем сбора устройств ввода/вывода на узлах ввода/вывода в каждой стойке. Мы сократили число прерываний системы перенеся на эти узлы ввода/вывода процессов ввода/вывода статистики многих вычислительных узлов.

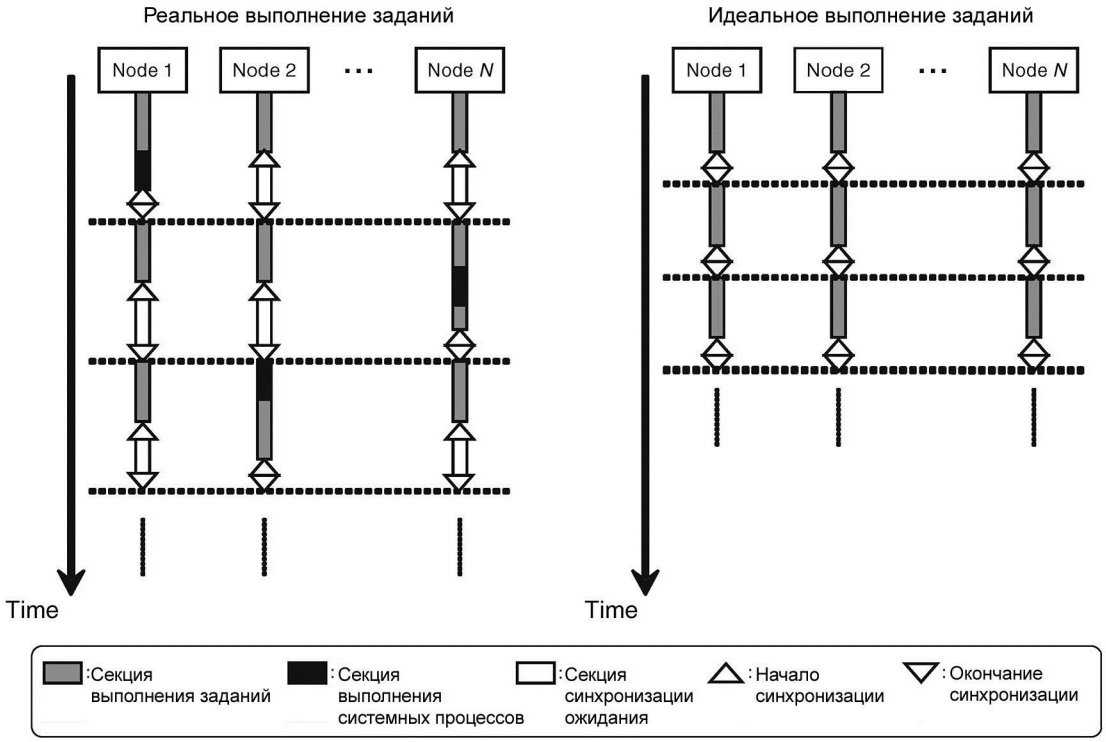

Рисунок 3
Работа планировщика синхронизаций

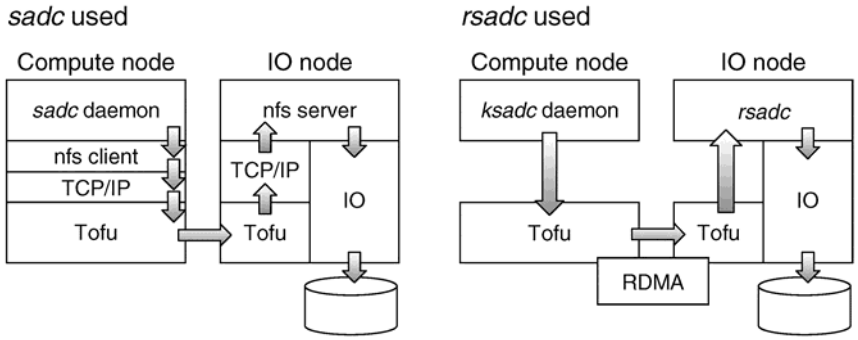

Рисунок 4
Схематическая диаграмма *rsadc*

Демон *ksadc* вычислительных узлов сохраняет статистику производительности отдельных узлов в памяти и затем осуществляет процессы ввода/вывода на узлах ввода/вывода с помощью *rsadc*.Утилита *rsadc* использует функцию прямого доступа к памяти (RDMA) интерконнекта Tofu для пересылки статистики производительности отдельных вычислительных узлов в память узла ввода/вывода без участия демонов на вычислительных узлах с последующим ее сохранением на диски.

Для К компьютера мы разработали функцию системы, состоящей из большого количества стоек, такую, что она может собирать файлы журналов (log), требующиеся для биллинга и управления системы с узлов ввода/вывода большого количества стоек без непосредственного доступа к вычислительным узлам. Это позволяет администратору получать понимание о состоянии всей системы не вызывая прерываний в работе параллельного задания.

5. Улучшение надежности

К компьютер, который состоит из большого количества аппаратных компонентов, является системой, от которой требуется различными пользователем и выполнение длительных вычислений, что нуждается в высокой надежности. В отношении аппаратных сбоев, например, важно иметь функции надежности, доступности и удобства использования (RAS: reliability, availability and serviceability), которые позволяют продолжать

выполнение работы даже при выходе из строя одной из компонент, а также раннее обнаружение точки сбоя для выполнения замены. Кроме того, система должна быть безопасной, обеспечивая разделение файлов внутри групп пользователей или предотвращая другим группам просмотр этих файлов.

К компьютер получил высокую надежность за счет использования аппаратного резервирования, функций RAS, имеющихся в Linux и включающих драйвера с множеством путей и функции безопасности, а также добавлением в Linux RAS-функций, усовершенствованных в существующих суперкомпьютерах и серверах Fujitsu.

В последующих разделах описываются примеры расширения функций RAS и безопасности.

5.1 Расширение RAS

ОС К компьютера унаследовала существующие серверные технологии Fujitsu для расширения функций восстановления после периодических ошибок памяти, а также идентификации и уведомлении о точках отказа аппаратных средств.

Для решения проблем ошибок памяти, вызванных альфа-лучами и тому подобным, ОС К компьютера сотрудничает с аппаратными средствами патруля памяти для выполнения коррекции данных, тем самым делая систему высоконадежной.

Аппаратные средства выполняют чтение (патрульное) памяти с интервалами, определяемыми операционной системой. С помощью механизма кода коррекции ошибок (ECC), ОС записывает данные в память в ней если обнаружена ошибка одиночного бита, над которыми выполняется коррекция ошибки, включающая ECC. Выполнение такой коррекции ошибок периодически предотвращает развитие корректируемых однобитных ошибок в некорректируемые двухбитные ошибки, что повышает надежность.

При обнаружении любой ошибочной страницы ОС помечает ее, предотвращая ее выделение при получении последующих запросов на выделение памяти, тем самым обходя эту страницу.

При обнаружении любой ошибки памяти, ОС помечает слот, содержащий эту память и уведомляет администратора о нем, тем самым сокращая время от остановки узла для замены сбойного компонента. Функции RAS были расширены чтобы позволить точную идентификацию возможных точек отказа основываясь на обнаруженном событии ошибки по отношению к CPU, устройствам ввода/вывода (канал PCI, адаптер Gbit Ethernet, адаптер Fibre Channel и т.д.) помимо памяти.

5.2 Функции безопасности

Система К компьютера используется одновременно многими пользователями и очень важно, что она имеет функции безопасности. Тем не менее, функции безопасности не имеют смысла, если они влияют на производительность вычислений.

Функции безопасности К компьютера были получены путем объединения функций управления пользователями и файловой системой обычно используемые в ОС, базирующихся на UNIX.

ОС поддерживает следующие функции безопасности при аутентификации системой общего управления пользователями

- шифрование паролей
- изменение прав доступа к файлам и директориям
- изменение владельцев фалов и директорий
- изменение групп хозяев файлов и директорий

Для безопасностью файлов и директорий управление осуществляется основываясь на управляемом профиле защиты доступа (CAPP, controlled access protection profile), который управляет доступом разрешением пользователям на "чтение", "запись" и "выполнение" для пользователя, группы и другого.

6. Заключение

В этой статье описаны особенности ОС К компьютера. Развитие ОС для беспрецедентной в мировом масштабе крупномасштабной системы было вызывающей задачей и мы должны были одновременно достичь и максимальную производительность, и удобство в использовании. Мы намерены непрерывно работать над развитием ОС для обеспечения основ проведения широкомасштабного моделирования на К компьютере и способствовать продвижению различных отраслей в науке и промышленности.