УДК 681.322

СУПЕРКОМПЬЮТЕРНЫЕ КЛАСТЕРНЫЕ СИСТЕМЫ – ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА

В.Н. Коваль, С.Г. Рябчун, И.В. Сергиенко, А.А. Якуба

Институт кибернетики им. В.М. Глушкова НАН Украины
03680, Киев, проспект Академика Глушкова,40,
тел. (044) 526-70-85; факс 526-45-49;
e-mail: icybcluster@gmail.com

В работе описываются основные цели первого в Украине суперкомпьютерного кластерного проекта СКИТ, который был спроектирован и построен в Інституте кибернетики ім. В.М.Глушкова НАН Украины, основы для выбора базовых решений в аппаратуре и программном обеспечении, характеристики кластеров. Работа содержит анализ достигнутой пропускной способности и описание способов её получения.

The paper describes main goals of SCIT - the first supercomputer cluster project in Ukraine built in Glushkov Institute of Cybernetics NAS of Ukraine, foundations for the designing its hardware and software and cluster characteristics. The paper contains the analysis of the performance results received on systems that were built and means used to do that.

Введение

В 2004-2005 гг. в Институте кибернетики им. В.М. Глушкова НАН Украины созданы и введены в исследовательскую эксплуатацию две высокоэффективные вычислительные системы – суперкомпьютеры с кластерной архитектурой СКИТ-1 и СКИТ-2 на базе современных микропроцессоров фирмы Intel. По своим характеристиками они не уступают мировым аналогам, а по возможности эффективной индивидуальной обработки крупных объемов знаний и данных в ряде случаев существенно превышают современные зарубежные образцы компьютерной техники.

Разработанные суперкомпьютеры СКИТ-1 и СКИТ-2 позволяют решать принципиально новые сверхсложные задачи большой размерности в области науки, экономики, экологии, сельского хозяйства, техники, в космической отрасли и других отраслях. Уже сегодня на семействе кластерных суперЭВМ реализован ряд прикладных пакетов для создания информационных технологий (ИТ) решения важных классов задач практического применения. В частности, в Институте кибернетики им. В.М. Глушкова НАН Украины разработаны программные пакеты для кластеров СКИТ-1 и СКИТ-2:

1) ИТ анализа устоявшегося движения жидкости в природных трехмерных многокомпонентных грунтовых объектах с полным или частичным влагонасыщением;

2) ИТ оптимального расположения сервисных центров (небольшое число поставщиков для обслуживания большого количества потребителей);

3) ИТ глубинной миграции дуплексных волн (на основе трехмерного полноволнового моделирования данных сейсмической разведки земной коры);

4) ИТ оптимизации последовательного обслуживания нескольких динамичных объектов в ситуации конфликта и неопределенности;

5) ИТ для матрично-векторных операций и решения систем линейных алгебраических уравнений;

6) ИТ автоматической морфологической разметки украиноязычных текстов с применением словарной базы данных (в рамках Национального корпуса украинского языка).

Даже этот ограниченный перечень интеллектуальных ИТ, которые прогрессируют с каждым годом, позволяет разработать новые механизмы и изучить производительность реализованных инструментов организации вычислений в кластерных системах семейства СКИТов. Итак, создание СКИТ-семейства кластеров, сгруппировав исследователей и объединив их усилия, стало не только инструментом решения сверхсложных задач супербольшого объема, но привело к замыканию обратной связи, когда преодоление проблем программирования и организации эффективных вычислений существенно способствует развитию самих интеллектуальных ИТ, способных решать все более сложные и сложные задачи.

Кроме Института кибернетики, сейчас СКИТ-семейство кластеров в режиме удаленного доступа (через Интернет) используют как совместный суперкомпьютерный вычислительный центр ряд институтов:

- Институт космических исследований НАН Украины и НКА Украины;
- Институт программных систем НАН Украины;
- Институт проблем математических машин и систем НАН Украины;
- Международный научно-учебный Центр информационных технологий и систем НАН Украины и МОН Украины;
- Институт металлофизики НАН Украины;

© A.Ju. Shelestov, N.N. Kussul, S.V. Skakun, 2006

- Институт молекулярной биологии и генетики НАН Украины;
- Институт сорбции и проблем эндоэкологии НАН Украины;
- Физико-технический институт низких температур НАН Украины (г. Харьков);
- Институт радиофизики и электроники НАН Украины (г. Харьков);
- Институт физиологии Киевского Национального Университета.

Для современных кластерных систем с сотнями процессоров (узлов) проблемы, решение которых желательно рассматривать как важнейшие тенденции организации распараллеливания вычислений и программирования, могут быть сформулированы следующим образом:

1. Наличие природного параллелизма классов решаемых задач и применения автоматических механизмов его учета, позволяющих существенно снизить время создания соответствующего программного продукта для кластерной системы.

2. Выяснение минимаксного объема задействованного процессорного ресурса кластерной системы для балансирования оптимальной нагрузки узлов кластера во время решения классов задач.

3. Существенное улучшение условий решения классов задач за счет применения разновидностей архитектуры и топологии соединений узлов кластерной системы, что дает возможность экономить ресурсы, прежде всего временные (сроки вычисления) и информационные (объем задействованной оперативной памяти или долгосрочных хранилищ). К тому же, спланированное изучение исследований в этом направлении позволяет снизить скорость морального устаревания кластерной системы и стоимость перманентного возобновления ее аппаратной составляющей.

Однако, сегодня говорить о конструктивной реализации перечисленных проблем рано по следующим причинам:

- К сожалению, нет ни одного продукта, который может претендовать на классификацию “средство автоматического распараллеливания”. Более-менее работающий вариант – openmp, да и он не автоматический, нужны директивы компилятору и поддержка openmp самим компилятором, к тому же, это распараллеливание работает только для SMP и MPP архитектур. Вряд ли автоматически можно что-либо распараллелить, кроме циклов, без участия программиста и при этом получить действительно приемлемую масштабируемость.
- Что касается балансирования оптимальной нагрузки, таким средством вполне может быть демон-планировщик со своим микроядром внутри, принимающий на себя небольшие фрагменты задач при наличии свободных ресурсов. Но сейчас в распоряжении разработчиков нет ни такого демона, ни инструментов для создания таких задач, хотя тема утилизации неиспользуемых ресурсов очень актуальна и интересна.
- На настоящий момент кластерные ресурсы можно экономить только в одном случае – если кластер создается под конкретные задачи с учетом всех особенностей этих задач, потребностей в памяти, интерконнектах, гигафлопсах, терабайтах, гигабайтах в секунду, допустимым временем исполнения с надёжностью 100 % и т.д. Только в этом случае можно просчитать необходимое количество узлов в кластере, количество процессоров и памяти на узел, выбрать используемый интерконнект, определить используемую распределённую файловую систему, объём системы хранения данных, количество необходимых для неё серверов, интерфейс для доступа узлов к системам хранения данных и т.д. Как только нужно создать кластер «на все случаи жизни», так сразу получаем разбалансировку по пропускной способности – либо процессоры слабоваты, либо их мало (или много, но интерконнект такое количество «не тянет»), и т.п. И самое главное, за такой кластер гарантированно переплачиваем. Снизить скорость морального устаревания могут только задачи: работающий кластер – не устаревший кластер. И чем точнее задача учитывает специфику конкретного или конкретных кластеров, тем лучше она будет работать и тем дольше оборудование не будет устаревать. Снизить скорость морального старения поможет объединение кластеров в *грид*, процессорных ресурсов всегда будет недостаточно и наличие рядом простаивающих, пусть несколько более медленных, ресурсов заставит пользователя их загрузить.

Вопросы, касающиеся идеологии разработки кластеров СКИТ-1 и СКИТ-2, их программно-аппаратного обеспечения, реализованных на них информационных технологий рассматривались в [1–3]. В 2005 г. была проведена существенная модернизация суперЭВМ СКИТ-1 и системы кластеров СКИТ-1 и СКИТ-2 в целом в части создания общей для всех серверов кластеров распределённой файловой системы путём объединения существующих систем хранения данных (СХД) в общую СХД, объединения функций сетей обмена данными и управления, и др. Сделанные изменения, и что не менее важно, опыт, полученный при решении задач различного типа и сложности, заставил разработчиков по-иному посмотреть на организацию совместной работы кластеров СКИТ-1 и СКИТ-2 и, соответственно, на организацию вычислительного процесса в них. Собственно, этому и посвящена данная работа.

1. Характеристики оборудования и системного программного обеспечения

Каждый из кластеров СКИТ-1 и СКИТ-2 представляет собой массив высокопроизводительных вычислительных узлов, управляющий узел, файловый сервер и локальные вычислительные сети. Обеспечивается удаленный доступ пользователя к кластеру, динамическая реконфигурация связи под задачу пользователя, осуществляется монополизация процессоров выделенных узлов, независимый от центрального управления обмен данными с другими узлами (рис. 1).

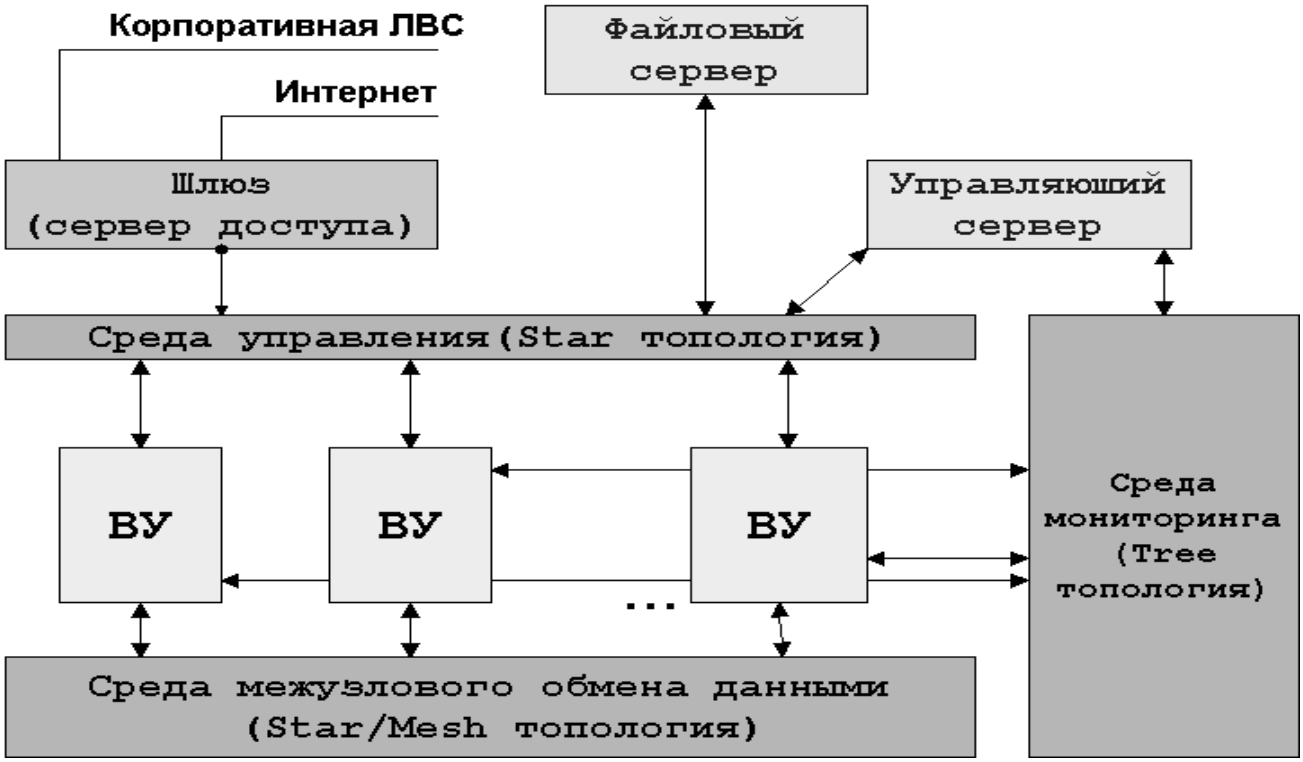

Рис. 1. Структура кластера

Управляющий узел кластера обеспечивает следующие функции: компиляцию задач пользователей, управление кластерными ресурсами, глобальное управление процессами на узлах кластера, файловое обслуживание задач, мониторинг кластерных объектов (узлов, задач, процессов).

В кластерной системе выделяется три среды передачи данных:

- среда управления – используется для управления узлами кластера, пропускная способность в диапазоне 100 MBps – 1 Gbps;
- файловая среда, используется для доступа узлов к общим файлам, пропускная способность в диапазоне 100 Mbps – 10 Gbps, примерами могут быть ***NFS***, или ***GFS*** или другая глобальная файловая система, например, ***Lustre,*** файловая среда может быть объединена с средой управления;
- среда вычислений (среда межузлового обмена данными) – в простейшем случае может использоваться Ethernet и она даже может быть объединена с файловой средой – используется для обмена сообщениями между вычислительными узлами, характеризуется двумя параметрами – латентность (чем меньше, тем лучше), и скорость (чем выше, тем лучше).

Каждая задача получает при исполнении запрошенный ею кластерный ресурс – он обычно измеряется в числе процессоров (а также и в числе узлов, их содержащих), во времени решения задачи (могут быть задачи очень длительного решения – недели счета), объем локального дискового пространства, если локальный дисковый ресурс имеется в кластере.

Характеристики оборудования СКИТ-1 и СКИТ-2 приведены в табл. 1, а состав общесистемного программного обеспечения – в табл. 2.

Длительное непрерывное решение задачи сопряжено с возможностью не получить результаты вовремя (например, из-за сбоев или при большой загрузке другими задачами), поэтому задачам предоставляется двойной временной ресурс – как полное время решения задачи, так и время непрерывного решения, после чего должна формироваться контрольная точка. Технология программирования с контрольными точками позволяет многократно прерывать исполнение и возобновлять его на промежуточных результатах.

Система управления задачами построена как традиционная клиент-серверная система. На клиентской стороне, которую в кластере представляет шлюз – сервер доступа, располагаются, в основном, домашние каталоги пользователей, доступ к которым каждый из них имеет полный, и из которого имеется возможность запускать задачу на узлы кластера и отслеживать её исполнение, если необходимо. Доступ вне собственного каталога, на управляющий сервер, узлы или даже домашние каталоги других пользователей, невозможен.

Связь клиентской части (access server) с серверной частью (control node) обеспечивается через программные пары-коммуникаторы, работающие на клиентской части по требованию, а на серверной части как фоновый процесс. Серверная часть программной пары анализирует, какое действие необходимо выполнить по оператору интерфейса пользователя, и передает управление и исходные данные для выполнения соответствующему средству серверной стороны. После выполнения необходимые данные возвращаются на клиентскую сторону (рис. 2).

Характеристики оборудования кластеров СКИТ-1 и СКИТ-2 Таблица 1

	СКИТ-1	**СКИТ-2**
Число узлов	**24**	**32**
Число вычислительных процессоров в кластере	**48 (Xeon 2,67 ГГц)**	**64 (Itanium2 1,4 ГГц)**
Число управляющих процессоров	**1**	**1**
Процессорный кэш	**1 Мбайт на процессор**	**3 Мбайта на процессор**
Всего ОЗУ (Гбайт)	**48 (DDR SDRAM PC-2100 ECC)**	**64 (DDR SDRAM PC-2100 ECC)**
Шины PCI	**2xPCI-X 100 Мгц, 1xPCI-X 133 Мгц**	**2xPCI-X 100 Мгц, 1xPCI-X 133 Мгц**
Связь межузловая	**Infiniband**	**SCI (Scalable Coherent Interface)**
Связь передачи файлов	**Gigabit Ethernet**	**Gigabit Ethernet**
Связь внешняя	**Fast Ethernet**	**Fast Ethernet**
Система хранения данных (Тбайт)	**1.6 (Общая для обоих кластеров)**	

Характеристики общесистемного программного обеспечения Таблица 2

	СКИТ-1	**СКИТ-2**
Операционные системы Linux	**Fedora Core 4, CentOS 4.2**	**Fedora Core 4**, **CentOS 4.2**
Ядро Linux	**2.6.12**	**2.6.12**
Файловая система	**Lustre 1.4.5**	**Lustre 1.4.5**
Системы параллельного программирования	**MVAPICH,Open MPI, LAM/MPI**	**Open MPI, Scali, LAM/MPI**
Языки программирования	**C, C++, Fortran-77**	**C, C++, Fortran-77**
Библиотеки	**ATLAS, FFTW, MKL, scaLAPACK**	**ATLAS, FFTW, MKL**
Пакеты моделирования	**Gromacs, Gamess,Wien2k**	**Gromacs**

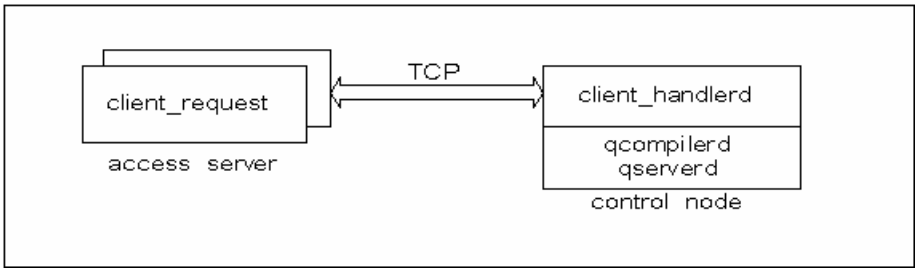

Рис. 2. Клиент-серверная схема интерфейса пользователя

В качестве операционной системы для кластеров были выбраны Linux FedoraCore4 и CentOS 4.2. Они установлены как на вычислительных, так и на серверных узлах кластеров. В качестве опорной (корневой) файловой системой для вычислительных узлов используется NFS, а для распределённой файловой системы выбрана Lustre [4].

Кластерное программное обеспечение, доступное пользователю, включает для языков программирования C/C++, Fortran компиляторы GNU и Intel различных версий, оптимизированные библиотеки для параллельных вычислений: ATLAS [5], BLACS [6], ScaLAPACK [7], Intel MKL [8], пакеты прикладных программ GROMACS [9], WIEN2K [10], GAMESS [11] и другие. В качестве параллельного интерфейса используются различные реализации MPI, как ScaMPI [12] и MVAPICH [13], «заточенные» для работы с конкретным интерконнектом, в нашем случае это SCI и InfiniBand, так и LAM/MPI [14]. OpenMPI [15], работающий сейчас в тестовом режиме и рассматриваемый нами в качестве замены LAM/MPI и MVAPICH, показывает неплохие результаты. При необходимости в систему могут быть добавлены как другие интерфейсные библиотеки, типа PVM, так и расчётные библиотеки, затребованные пользователем.

2. Характеристики потока задач и их влияние на структуру

Источниками задач для кластерного комплекса являются:

- локальный доступ к кластеру (исключительный случай, для однократного ввода-вывода гигабайтных данных);
- корпоративная ЛВС Института кибернетики (может обслуживать пользователей через местную АТС со скоростями 30 – 1000 Kbps;
- обычный Интернет (скорость до 256 Kbps);
- академическая оптоволоконная ЛВС (со скоростями гигабитного диапазона).

Последний вид связи находится пока в предстартовом состоянии, но только через него можно организовать удаленное решение задач с большими объемами данных. Для остальных источников соотношение по объемам данных изменялось от 10:90:0 в начале эксплуатации кластеров до сегодняшнего 10:40:50 и эта тенденция будет усиливаться по мере перехода к широкополосной ЛВС, а в будущем и к широкополосному Интернету.

В самом начале эксплуатации кластеров сведения о специфике и параметрах задач, предназначенных для исполнения на кластерах, были весьма приблизительны, достаточно сказать, что некоторые базовые параметры – объем данных в десятки Мбайт и время решения задачи в часы – оценивались как присущие большим задачам. Соответственно, на такие параметры была рассчитана и стартовая организация вычислительного процесса (внутрикластерная сеть управления с пропускной способностью до 1 Gbps, простая файловая система хранения данных на основе RAID-0,1,5, файловая система NFS как основное средство взаимодействия компонент кластерного комплекса, обслуживаемая по сети управления, все пользовательские данные расположены только в его домашнем каталоге).

Реальность в действительности оказалась совсем иной: наравне с отладочными задачами, которые могут занимать все процессорные ресурсы, но на короткое время (до 20–60 мин), и обычно ограничены только этими ресурсами, а также наличными объемами оперативной памяти, появились задачи, которые очень плохо “укладываются” в стартовые ограничения вычислительного процесса в кластерах. Это несоответствие проявлялось, в основном, в файловой работе и в размерах файлов данных, подлежащих обработке, а также в тех системных затратах, которые сопровождают доставку этих данных из домашнего каталога к месту их обработки на управляющем сервере и на узлах кластера. *Если для отладочной задачи с небольшим объемом исходных данных принятая в кластерах технология копирования стартового каталога задачи в рабочий каталог исполнения на управляющем сервере проходила незаметно, то для исходных данных в десятки гигабайт копирование этих объемов и возможный возврат их в домашний каталог при неудачной компиляции создает недопустимую нагрузку на сетевые пересылки* (*к тому же часть из них – возврат обратно неизмененных данных – являются совершенно лишними*).

С другой стороны, на первое место среди архитектурных дефектов кластеров выходит *несбалансированность* пропускных способностей разных частей кластера – если пропускная способность интерконнекта в 250–800 Мбайт/с практически никогда не выбирается до конца, то 120 Мбайт/с гигабитного канала управления явно недостаточно для интенсивной файловой работы. В дополнение к последнему, недостаточно создать простую файловую систему, индивидуальную на каждом кластере, *кластерный комплекс требует единой файловой системы*, а ее выбор должен быть осознанно ограничен только теми, которые специализированы как *глобальные* файловые системы.

Все задачи, подлежащие решению на кластерах, т.е. задачи с существенным распараллеливанием вычислений, потребностью в большом объеме этих вычислений (возможно, дни и недели счета) и/или интенсивной работой с файлами суммарным объемом в десятки и сотни гигабайт, могут быть отнесены (условно) — к следующим типам:

- ограниченные только процессорными ресурсами или объемами оперативной памяти;
- ограниченные обменами по межузловым связям;
- ограниченные файловым обслуживанием.

Как оказалось, существующие для кластеров прикладные пакеты моделирования в областях молекулярной биологии или квантовой химии, относятся именно к последнему типу, что *при многосуточном непрерываемом счете каждой задачи всего несколько задач могут полностью занять все кластерные ресурсы на долгое время* (естественно, для таких условий, когда кластер имеет только небольшое число процессоров – несколько десятков или сотен). Существенным недостатком здесь является требование непрерываемого счета, т.е. работа с контрольными точками имеет очень ограниченные функциональные возможности, выходом из этой ситуации может быть переписывание пакета, что представляется нереальным, или капитальная реконструкция всего вычислительного процесса на кластере и балансирование характеристик кластеров по пропускной способности – для минимизации негативных последствий организации вычислений в таком пакете.

Балансирование предполагает следующий перечень изменений (по сложности реализации):

- увеличение пропускной способности гигабитных каналов за счет транковых передач;
- реализация режима “persistence data” для индивидуального пользователя (возможность хранения вне домашнего каталога и подключение к исполняемой задаче по требованию больших файлов данных);
- *выбор и инсталляция глобальной файловой системы (т.е. объединение дисковых ресурсов всех кластеров в единый ресурс при одновременном увеличении скорости выполнения файловых операций).*

3. Архитектура кластерного комплекса

Архитектура многосерверной, кластерной системы – это многоплановая комбинация аппаратно-программных средств, в том числе на уровне взаимодействия операционных систем серверов, распределения вычислительных процессов по процессорам и синхронизация этих процессов, эффективное обслуживание запросов к централизованным или распределенным файловым системам.

В этом разделе будут рассмотрены конкретные компоненты архитектуры, характеристики которых существенно влияют на организацию вычислительного процесса в кластерной системе.

Сети управления и обмена данными. Сеть обмена данными (СОД) в общем случае должна предоставлять возможности:

- удаленное включение вычислительного узла через протокол WakeOnLan;
- доступ узла к данным о сетевой конфигурации (протоколы DHCP);
- загрузку операционной системы в вычислительный узел (протоколы TFTP);
- доступ узла к корневой файловой системе (протоколы NFS);
- поставку в вычислительный узел данных задачи (протоколы NFS).

Сеть управления (СУ) в общем случае обеспечивает возможность доступа к узлу извне для:

- оперативного управления узлом;
- получение статистических данных по нагрузке процессоров, занятость памяти, показание датчиков температуры, скорости вращения вентиляторов;
- запуск и дальнейший контроль процессов задачи.

Рассмотрим подробно использование сети обмена данными вычислительным узлом кластера. Каждый узел сконфигурирован на включение при получении сетевым интерфейсом специального пакета **wake-on-lan** и на загрузку через сетевой интерфейс по протоколу PXE. Управляющий узел отправляет сформированный пакет через СОД и узел инициирует процесс загрузки (на самом деле пакет **wake-on-lan** может быть отправлен и через СУ, но это увеличивает объем конфигурационной информации в два раза, а, значит, и возможность ошибки возрастает).

Узел отправляет широковещательный запрос и от сервера DHCP, который установлен на управляющем узле, получает всю необходимую для загрузки системы информацию, загружает ядро и минимальную корневую файловую систему с сервера TFTP, который также установлен на управляющем узле, распаковывает ядро и передает ему выполнение.

Дальше процесс инициализации системы, базируясь на полученную по DHCP информацию, монтирует по NFS файловую систему, расположенную на СХД, делает ее корневой и завершает инициализацию, передав управление стартовым скриптам, расположенным на новой корневой файловой системе. Собственно, с этого момента загрузка системы по сети или с локального диска практически не отличается. В дальнейшем монтируются дополнительные разделы NFS с рабочими данными, пользовательскими каталогами и т.д. по необходимости.

Такая схема, предполагающая, что *корневая файловая системы у всех узлов кластера одна и та же, существенно облегчает администрирование, обновление, установку нового программного обеспечения*, поскольку работает со всем кластером целиком, и на порядки снижает возможность совершить ошибку. Корневые файловые системы всех узлов идентичны, за исключением нескольких каталогов, которые действительно должны быть у каждого уникальны. Процессы каждой запущенной задачи, работая каждый на отдельном узле, все равно работают в одном и том же каталоге, расположенном в СХД, читают и пишут из/в одни и те же файлы.

Как видим, практически вся работа по файловому вводу-выводу производится по сети обмена данными, поэтому требования к пропускной способности этой сети очень высокие. Следует уточнить, что «бутылочным горлышком» в этой схеме является сетевой интерфейс сервера, пропускной способности сетевого интерфейса узла хватает с избытком.

Запуск задачи на исполнение на узлах кластера может быть осуществлён различными способами в зависимости от задачи, т.е. MPI-задача запускается командой *mpirun,* а обычная не параллельная программа может быть запущена командой *ssh* либо *rexec*. Для оперативного контроля за состоянием запущенной задачи, для ее принудительного завершения, и освобождения занятых задачей ресурсов также используется доступ к узлу по протоколу ssh. Это подразумевает, что узел должен быть доступен.

Изначально планировалось использовать две совершенно разные сети – в соответствии с хорошо известными более 10 лет подходами к построению кластеров. Сеть обмена данными была построена на интерфейсах Gigabit Ethernet (1000 Mbps) и управляемом сетевом оборудовании, а сеть управления на интерфейсах Fast Ethernet (100 Mbps) и неуправляемом сетевом оборудовании. Это позволяет отделить файловый доступ от управления узлами даже при стопроцентной загруженности сети.

Однако в ходе эксплуатации кластеров оказалось, что *при максимально достижимой загрузке сети обмена данными стопроцентная утилизация сетевых интерфейсов не достигается и в нашем распоряжении всё равно остается запас по пропускной способности, достаточный для сети управления*. Видимо, упомянутые подходы базировались на использовании только Fast Ethernet в качестве сети обмена данными и не учитывали возросшую на порядок ширину полосы пропускания. *Поэтому сейчас реализован полный отказ от*

выделенной сети управления, а функции сетей обмена данными и управления объединены. Бонусом такого решения есть уменьшение в два раза количества сетевых кабелей, а значит и уменьшение возможных точек отказа, т.е. повышение общей надежности сетевой инфраструктуры кластеров.

Однако необходимость в базовом удалённом управлении каждым узлом кластера в отдельности, возможность выполнения таких операций как включение-выключение узла, консоль с выводом загрузки узла потребовала установки **ServNET** [16]. В дальнейшем планируется использовать узлы только с поддержкой стандарта **IPMI** [17] версии выше 1.5, обеспечивающего удалённое включение-выключение узла при наличии только подключенного к узлу ethernet кабеля и питания, а функция Serial-over-LAN в IPMI 2.0+ позволяет даже удалённо настраивать BIOS узла.

В дальнейшем планируется провести эксперименты по переносу функции доступа к файлам на вычислительную сеть. Суть идеи заключается в том, что вычислительная сеть, построенная на интерфейсах Infiniband, избыточна по пропускной способности и в 10–60 раз превосходит по этому показателю Gigabit Ethernet, что позволяет использовать в качестве сети обмена данными только часть вычислительной сети без падения производительности и пропускной способности последней. При подтвержденных экспериментами хороших результатах это позволит увеличить не только скорость чтения/записи файлов, но и за счет пониженной латентности такой важный для кластерных баз данных показатель, как скорость позиционирования.

IP-сеть. В качестве опорной в кластерах используется IP-сеть, при этом следует учитывать следующие ограничения и пожелания:

1. Ограничение только приватными диапазонами IP-адресов: 10.0.0.0 – 10.255.255.255, 172.16.0.0 – 172.31.255.255, 192.168.0.0 – 192.168.255.255.
2. Каждый кластер должен использовать свою собственную подсеть ip-адресов.
3. IP-адрес должен быть максимально информативным.
4. Управляющим может быть как выделенный узел, так и невыделенный, т.е. один из вычислительных узлов кластера.
5. Системы хранения данных должны быть доступны во всех кластерах без дополнительной маршрутизации через промежуточные узлы.
6. Система хранения данных может быть также и управляющим узлом кластера.

Как видим, п. 2 и 5 противоречат друг другу, в тоже же время они необходимы. Для разрешения противоречия может быть применен протокол VLAN на сетевых коммутаторах, позволяющий на канальном уровне ограничить доступность портов других кластеров.

В результате выбран был приватный диапазон 10.0.0.0 – 10.254.254.254 как наиболее просторный, и в нем применена следующая схема распределения подсетей IP-адресов:

Вычислительный узел имеет IP-адрес 10.N.M.X, где N – номер кластера, M – номер коммутатора, X – номер порта в коммутаторе. Таким образом, 10.1.1.1 – это первый узел первого кластера, а 10.3.1.24 – двадцать четвертый узел третьего кластера.

Маска IP-адреса 255.0.0.0, т.е. вся сетевая инфраструктура полностью достижима из любой точки, при этом разграничение различных кластеров выполняется с помощью VLAN-ов. В результате узлы различных кластеров взаимно невидимы, при этом системы хранения данных будут доступны даже в случае, если функции СХД и управляющего узла кластера возложены на одно устройство.

Коммутаторы кластера имеют фиксированные IP-адреса: 10.N.M.250, где N – номер кластера, M – номер коммутатора.

Управляющий узел кластера имеет фиксированный IP-адрес: 10.N.M.254, где N – номер кластера, M – номер коммутатора.

Подсеть 10.0.0.0/16 отдана для сервисных служб, так 10.0.0.254 – это сервер доступа, 10.0.1.0/24 – устройства бесперебойного питания и т.д.

Для более упрощенного наименования узлов кластера используется DNS сервер со следующей схемой наименования: nXXX.cNN.icyb, где XXX – номер узла в кластере, NN – номер кластера. Таким образом, узлы кластера СКИТ-2 имеют имена n001.c03.icyb – n032.c03.icyb (имя управляющего сервера этого кластера – n000.c03.icyb).

Файловый сервис. Как правило, параллельные задачи ориентированы на вычисления, связанные с огромными массивами начальных, промежуточных или конечных данных. *Так, анализ результатов ядерных исследований может использовать сотни терабайт исходных данных, а задача в пакете квантовой химии Gamess* [11] *создает временные файлы размером в несколько гигабайт на процесс с постоянным чтением-записью в них промежуточных результатов мелкими пакетами*. Поэтому крайне важная задача – предоставить узлам высокоскоростной доступ к системам хранения данных огромных размеров.

До декабря 2005 г. в качестве систем хранения данных выступали управляющие узлы двух кластеров. Для увеличения пропускной способности системы хранения данных используется PORT TRUNKING – объединение 2-4 сетевых интерфейсов в один с увеличением общей пропускной способности полученного интерфейса, хотя линейного прироста получить не удалось. (При использовании обычного транкинга IP-стек OS Linux имеет ошибку обработки пакетов при дефрагментации, что может приводить к потерям пакетов до 30 %, ошибкам на интерфейсах и, как результат, к меньшему ожидаемому приросту. Вариантов решения проблемы два: воспользоваться протоколом для транкинга 802.3ad при поддержке его оборудованием и/или

воспользоваться JUMBO FRAMES. К сожалению, на оборудовании, имеющемся на наших кластерах, мы не смогли получить ни то, ни другое. Возможно, ситуация будет исправлена с выходом новых прошивок для коммутаторов). СХД на СКИТ-1 имеет объем 0.4 Тбайт, созданных на 4 SCSI-дисках в RAID5 с максимальной скоростью линейного чтения и записи в 80 Мбайт/с. СХД на СКИТ-2 имеет объем 0.8 Тбайт, созданный на двух двухканальных контроллерах и 26 SCSI-дисках в RAID50, с передачей данных по четырем сетевым интерфейсам с максимальной линейной скоростью чтения-записи в 330 Мбайт/с.

В качестве распределенной файловой системы использовалась NFS. Узлы кластера не имеют собственных дисков, поэтому каждый узел во время начальной загрузки монтирует корневую файловую систему по NFS. Также по NFS монтировались и разделы с рабочими данными задач.

Выбор NFS был обусловлен несколькими причинами — это стандартная сетевая файловая система, NFS имеется в любой UNIX-системе, NFS очень легко настраивается и конфигурируется.

Опыт эксплуатации NFS в течение года в качестве основной файловой системы показал, что NFS является отличным выбором для небольших (на 4-8 узлов) кластеров. Для кластеров уровня СКИТ-1 и СКИТ-2, на 16-32 узла, NFS может быть неплохим выбором при условии использования для счета задач с небольшим количеством операций ввода-вывода с дисковыми файлами. Однако NFS становится неприемлемым выбором при использовании для решения задач с интенсивным вводом-выводом. Так, при эксплуатации пакета Gamess загрузка NFS на СКИТ-1 достигла максимума, вплоть до возникновения отказа в обслуживании управляющего узла. Ситуацию исправил только перевод данных задачи на более производительную СХД СКИТ-2.

Поэтому файловый сервис был модернизирован, и основными задачами модернизации стали:

выбор наиболее оптимальной распределенной файловой системы с возможностью масштабирования как по объему с возможность объединить существующие СХД разных кластеров в одну общую СХД, так и по максимальной пропускной способности;

переход с использования NFS на частичное или полное использование выбранной файловой системы.

Рассматривались следующие кандидаты на роль распределенной файловой системы:

GFS производства RedHat (ранее SISTINA.COM) [18], на сегодня последняя версия 6.1. В качестве распределенного хранилища данных использует монтируемый одновременно всеми узлами GNBD (global network block device), поверх которого работает собственно GFS со своим менеджером блокировок.

Достоинств у GFS довольно много – это бесплатность решения, разработка крупнейшим производителем RedHat Linux, работа «прямо из коробки» при использовании RedHat Enterprise Linux 4 и выше или Fedora Core 4 и выше, неплохая масштабируемость по объему, легкость в установке и конфигурировании.

В тоже время есть и недостатки – плохая масштабируемость по общей пропускной способности, а это значит, что для наращивания емкости придется использовать дорогостоящие аппаратные решения типа FiberChannel, поскольку все узлы разделяет одно блочное устройство, то отказ одного из узлов может привести к некоторым повреждениям файловой системы.

GFS является неплохим выбором для законченного решения, когда не планируется наращивать вычислительную мощность кластера и объем системы хранения данных, т.е. поставка под ключ. Для использования в нашем случае, предполагающем дальнейшее наращивание мощности по вычислительным ресурсам и объемам дискового пространства, мало подходит.

Производства IBM [19], наследник с открытыми кодами AFS, которой уже исполнилось 25 лет, последняя версия 1.4.0. Достаточно надежная, очень быстрая в правильной конфигурации файловая система.

Для высокой скорости требуется наличие в узле жесткого диска для локального кеширования данных, а сама система ориентирована на преимущественное чтение данных. Высокая скорость достигается за счет большого количества серверов хранения данных и кеша со стороны клиента. Вполне выдерживает сотни терабайт одновременно открытых файлов и 50000 одновременных подключенных узлов.

На небольшом количестве СХД (2–8) может проиграть NFS. Минимальное количество СХД – 2. Чаще всего используется для громадных кластеров, как правило, производства IBM, или громадных FTP-серверов. OpenAFS – бесплатная, но очень сложная в установке и конфигурировании.

По двум причинам OpenAFS мало подходит для использования в качестве распределенной файловой системы в наших системах – мы не скоро достигнем уровня по количеству СХД, когда общая производительность выйдет на приемлемый уровень, и мы не используем локальных дисков на узлах (по меньшей мере сейчас), а, следовательно, не получим выигрыш в скорости за счет кеширования операций ввода-вывода.

OCFS2 производства ORACLE [20], наследник с открытыми кодами OCFS. Пока стабильной версии системы еще нет, но она уже есть в ядре Linux, поддерживается дистрибутивами Linux RHEL4, SLES9, UBUNTU, DEBIAN. Фактически представляет собой распределенный по всем узлам кластера RAID5-массив, что дает как высокую скорость чтения-записи, так и некоторую отказоустойчивость всего массива. Однако, если несколько узлов вышли из строя, то массив может развалиться вплоть до потери всех данных, т.е. система требует высоконадежной дисковой подсистемы на каждом узле кластера, что очень сильно увеличивает общую цену решения. OCFS2 оптимальна для обработки больших баз данных, для чего собственно и создавалась.

Для применения в наших условиях не подходит, поскольку если OCFS2 будет установлена на СХД, то объединенные данные должны будут экспортироваться с помощью все той же NFS (т.е. мы возвращаемся к конфигурации, от которой хотим уйти, с той лишь разницей, что теперь у нас будет объединенная файловая

система), а для разнесения данных по всех узлам кластеров необходимо спроектировать архитектуру кластера заново с дополнительными, и немалыми, финансовыми затратами.

Lustre производства CLUSTERFS.COM [4], коммерческая, бесплатная версия, выходит с некоторым отставанием, максимально на год. Очень сложна в установке, но очень проста в конфигурировании. Отлично масштабируется как по объему, так и по пропускной способности.

Достаточно высокие требования по надежности предъявляются к самим СХД. Но поскольку наши технические условия вполне соответствовали требованиям к Lustre, эта файловая система была выбрана как основной кандидат на роль распределенной файловой для кластеров, тем более, что с декабря 2005 г. в clusterfs.com полностью изменена система лицензирования, теперь продукт доступен для бесплатного использования с момента появления.

В декабре 2005 – январе 2006 гг. кластеры СКИТ-1 и СКИТ-2 были переведены на использование распределённой файловой системы Lustre. Это позволило объединить все СХД кластеров в одну общую файловую систему объемом 1.7 Тбайт. Физически общая файловая система располагается на трёх серверах данных (OSS) с четырьмя дисковыми массивами (OSD) и одном сервере метаданных (MDS). Поскольку при конфигурировании Lustre мы указали распределять файл по всем OSD (фактически это классический RAID-0 в применении к файлу), то таким образом мы смогли распределить нагрузку по файловому вводу-выводу одновременно на все серверы.

Результаты тестирования двух файловых систем, Lustre и NFS, на файле размером в 8 Гбайт (в тестировании измеряются: пропускная способность – Кбайт/с, использование процессора, частота поиска) приведены в табл. 3.

Таблица 3

Operation	Sequential Output						Sequential Input					
	Per Char		Block		Rewrite		Per Char		Block		Random seek	
	Kbps	%CPU	Kbps	%CPU	Kbps	%CPU	Kbps	%CPU	Kbps	%CPU	k/sec	%CPU
NFS	**26665**	**88.1**	**27907**	**6.3**	**3134**	**95.2**	**29215**	**91.1**	**84975**	**15.3**	**460.7**	**2.8**
Lustre	**27791**	**99.3**	**69991**	**41.3**	**39668**	**58.0**	**28066**	**98.9**	**98254**	**86.1**	**121.6**	**17.3**

В дальнейшем можно нарастить объем и пропускную способность файлового хранилища путём простого подключения к коммутатору дополнительных серверов с дисковыми массивами и небольшого переконфигурирования самой системы.

4. Организация безопасности кластерного комплекса

Доступ пользователей в систему. Имеется два аспекта безопасности кластерного комплекса – во-первых, защита самого комплекса от несанкционированного доступа (в том числе и от зарегистрированных пользователей) и, во-вторых, каждому из пользователей должна быть гарантирована индивидуальная защита его домашнего каталога от других пользователей.

Защита сервера доступа осуществляется функциями брандмауэра. Тема настройки брандмауэра на Linux здесь рассматриваться не будет, просто уточним, что коммуникация с сервером доступа допускается только по защищенным протоколам.

Регистрация пользователя проводится на LDAP-сервере и доступна по LDAP-протоколу в любой точке кластера (ранее регистрация производилась на сервере доступа в локальной системе безопасности). Для увеличения отказоустойчивости системы работают два резервных LDAP-сервера. Домашние каталоги пользователей располагаются на распределённой файловой системе кластера.

Защита пользователей основана на использовании защищённых протоколов для доступа к системе (SSH), криптостойких паролей входа в систему, RSA или DSA ключей и применении необходимых прав доступа и ACL (access control list) к пользовательским каталогам и файлам. Для пересылки данных между пользовательским компьютером и сервером доступа используются защищённые протоколы SSH (команда scp, sftp).

Перспективным решением является перевод системы доступа пользователей на Web-интерфейс с использованием протокола HTTPS. Этот протокол использует шифрование с помощью протоколов SSL (Secure Socket Layer), обеспечивающий достаточно высокий уровень защищённости канала передачи данных между кластером и пользователем. Таким образом, можно объединить защищённость канала и привычную среду работы с Web-платформой, которая на сегодняшний день считается одной из наиболее перспективных платформ.

Доступ к компонентам системы. Возможны несколько регламентов исполнения пользовательских задач:

- Зарегистрированному пользователю предоставляется право доступа на все вычислительные узлы, выделенные для задачи, после окончания которой, это право прекращает действие.
- Пользователь передает свою задачу при старте в управление псевдопользователю, а сам только может контролировать ход её выполнения и получать результаты как промежуточные, так и окончательные.

Ранее для кластеров СКИТ-1 и СКИТ-2 из соображений, как безопасности, так и по причине разношерстности систем хранения данных, был выбран второй вариант, хотя он и связан с существенным ограничением прав контроля поведения собственной задачи на узлах кластера со стороны пользователя.

Задача после постановки в очередь старта перемещалась в рабочий каталог (для каждой задачи индивидуальный) для исполнения, и выполнялась на выделенных ресурсах кластера уже средствами системы управления задачами (СУЗ) и правами одного из псевдопользователей, а после выполнения результаты возвращались пользователю.

Недостатки этого подхода очевидны – излишнее копирование данных, которые для некоторых задач могут быть весьма внушительны, потеря места на системах хранения данных и другие. Результат борьбы с проблемами производительности вылился в использование технологии Persistent Data (инструмент, позволяющий пользователю заранее поместить на постоянное хранение большие массивы данных, избавляя от длительного копирования при постановке задачи в очередь).

После внедрения Lustre в качестве основной файловой системы кластеров мы, расположив домашний каталог пользователя на распределённой файловой системе, автоматически решили проблему копирования больших массивов данных из пользовательского окружения в рабочий каталог задачи. Теперь рабочим каталогом задачи выступает каталог, заданный пользователем. Но это решение потребовало коренной переработки всей системы управления очередями задач, изменило все, применяемые нами ранее, подходы к безопасности выполнения задачи. Нам пришлось отказаться от использования псевдопользователей и дать возможность выполняться задаче с правами её владельца. С одной стороны, это решение серьёзно упростило всю цепочку управления очередями задач, с другой – система безопасности всё ещё находится в состоянии становления.

Безопасность аппаратуры кластерного комплекса. Кластерный комплекс Института кибернетики им. В.М. Глушкова НАН Украины функционирует в условиях нестабильного внешнего электропитания, а наличие резервного электропитания не предусматривалось по финансовым соображениям.

В соответствии с этим подходом изменилась структура кластерного комплекса – комплекс был разделен на 2 группы аппаратно-программного обеспечения, в первую группу были выделены все управляющие серверы, коммутаторы управления и система хранения данных – тем самым обеспечивается удаленное управление всеми компонентами кластерного комплекса по IP-адресам и доступ к хранимым данным в дисковых RAID-массивах. Во вторую группу отнесены все вычислительные узлы и интерконнект, причем в каждый сервер вычислительного узла вводится дополнительное аппаратно-программное средство, позволяющее удаленно включать – отключать, настраивать и контролировать каждый вычислительный узел индивидуально.

В состав программного обеспечения кластерного комплекса введены средства постоянного мониторинга, начинающие работать в регламентном режиме сразу после включения управляющих серверов и не создающие существенную нагрузку, как на управляющие серверы, так и на вычислительные узлы.

Упрощенная блок-схема кластерного комплекса, рассчитанного на круглосуточную эксплуатацию и удаленное управление, показана на рис. 3.

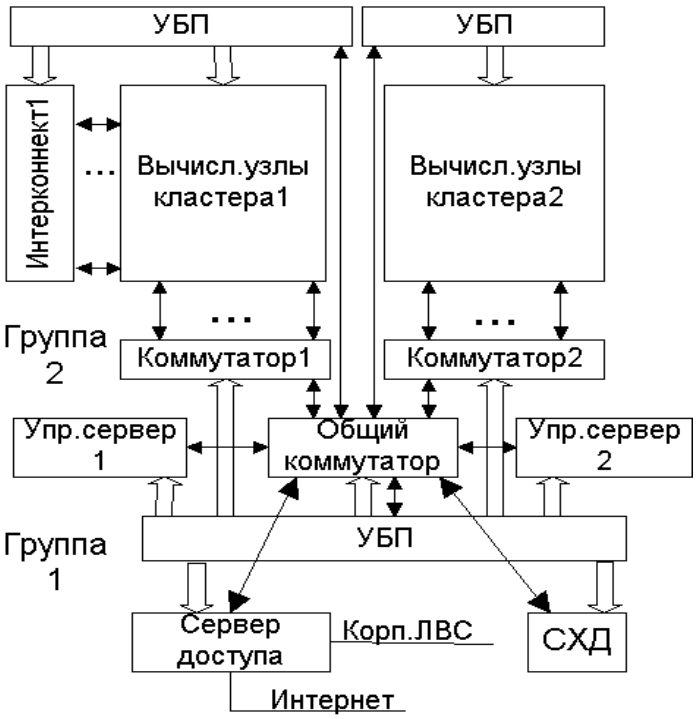

Рис. 3. Блок-схема кластерного комплекса

Здесь:

Группа 1 – устройства, выделенные в группу устройств с постоянным электропитанием, они обеспечивают удаленный доступ ко всем компонентам кластерного комплекса. В группу 1 входят:

- управляющие серверы 1 и 2 кластеров;
- сервер доступа;
- коммутаторы сети управления 1 и 2 кластеров;
- общий коммутатор кластерного комплекса;
- система хранения данных (СХД);
- устройство бесперебойного питания (УБП) на 5 кВт.

Группа 2 содержит устройства с большим электропотреблением при их работе. В группу 2 входят:

- вычислительные узлы 1 и 2 кластеров;
- коммутатор интерконнекта для кластера 1 (на InfiniBand'e);
- коммутаторы сети мониторинга на основе Fast Ethernet (на схеме не приведены);
- УБП по 10 кВт на каждый шкаф.

Связи инфраструктуры кластерного комплекса по электропитанию показаны как однонаправленные (от УБП к устройству-потребителю) широкие стрелки, связи по IP-адресам (а эта адресация охватывает все устройства кластерного комплекса) показана как двунаправленные стрелки или стрелки без указания направления.

Общий коммутатор в группе 1 может быть необходим по двум причинам:

– для обслуживания оптоволоконного кабеля, через который идет удаленное управление кластерным комплексом;

– для балансировки пропускной способности передач по протоколам NFS между компонентами комплекса, что выполняется созданием виртуальных подсетей и организацией транковых передач между ними.

5. Анализ производительности кластеров

Архитектурные решения и реальная производительность. Специфические свойства параллельной кластерной задачи:

- задача – это множество процессов, имеющих идентичный код, запущенных на разных узлах кластера и выполняющих часть общей работы;
- во время работы процессы могут производить интенсивный обмен данными между собою;
- межпроцессный обмен данными приводит к выравниванию производительности каждого процесса по скорости самого медленного;
- каждый процесс, как правило, требует во время исполнения большое количество оперативной памяти.

Исходя из этих свойств, можно сформулировать общие требования к узлу кластера:

- производительность узла напрямую зависит от мощности процессора;
- межпроцессорный обмен данными всегда быстрее межузлового обмена, т.е. предпочтительнее использовать многопроцессорные узлы (на 2–4 процессора) и многоядерные процессоры (сейчас процессоры с двумя ядрами, а в недалеком будущем и с 4 ядрами);

Производительность узла напрямую зависит от:

- частотных характеристик используемой шины оперативной памяти;
- количества доступной в узле оперативной памяти (до некоторого разумного предела);
- типа используемого интерконнекта, при этом важными являются две характеристики – латентность, т.е. задержка, возникающая при передаче минимального пакета между узлами, и максимальная пропускная способность;
- интенсивности операций ввода-вывода с устройствами хранения данных.

Рассмотрим более детально влияние свойств процессора и памяти.

Конвейер и системные вызовы. Как правило, параллельные задачи используют линейные алгоритмы, поэтому классическая архитектура с коротким конвейером, используемая в процессорах AMD, гораздо предпочтительнее архитектуры P4 процессоров INTEL. Каждое обращение к данным соседнего процесса сопровождается несколькими переходами в привилегированный режим процессора. Цена этого перехода на процессорах AMD 120-240 тактов, на процессорах архитектуры P4 1100-1300 тактов.

HyperThreading. За счет простоя одного из конвейеров при неверно предсказанном переходе или просто невозможности параллельного исполнения инструкции на архитектуре P4 есть возможность использования простаивающих ресурсов в качестве виртуального процессора (HyperThreading), но в параллельных задачах это приводит только к падению производительности. Причина проста – межузловой обмен выравнивает производительность всех процессов по скорости самого медленного и, поскольку на виртуальный процессор приходится не более 40 % реального процессора, то и общая производительность падает в 2–3 раза, т.е. эта возможность для кластеров практически бесполезна.

64 бита против 32 битов. На сегодня все процессоры либо поддерживают 64–битные расширения (AMD64, EM64T), либо являются чистыми 64-битными процессорами. К сожалению, сейчас выигрыш от использования разрядности в 64 бита получают только программы, нуждающиеся в вычислениях с такой

арифметикой, да и то не всегда, остальные только проигрывают. Причин этому несколько (из-за увеличенного вдвое размера данных и адреса):

- Требуется увеличить вдвое кеши процессора, иначе наблюдается падение производительности при частом «вымывании» кеша.
- При той же ширине шины памяти требуется вдвое большее количество обращений к оперативной памяти, что дает падение производительности.
- Требуется увеличения вдвое оперативной памяти узла.

Потребляемая мощность процессора. Выделяемая мощность процессора может неявно влиять на общую производительность всей системы – при перегреве одного из процессоров к нему будет применено автоматическое понижение частоты, что сразу же приведёт к общему падению производительности всей системы в целом.

Оперативная память. Кластеры имеют 1-2 Гбайта на каждое процессорное ядро узла:

- больше 2 Гбайт на процессорное ядро целесообразно либо при использовании чистой 64-битовой архитектуры, либо после уточнения специфики основных прикладных задач кластера, иначе память будет существенно простаивать;
- частота, на которой работает оперативная память, должна быть максимальной из всех поддерживаемых выбранной архитектурой процессора;
- используемый чипсет должен уметь поддерживать необходимое количество памяти.

Интерконнект. Поскольку цена интерконнекта лежит в весьма широком диапазоне от нуля до нескольких тысяч долларов на узел, то и выбор интерконнекта определяется основным назначением кластерной системы:

- Латентность интерконнекта – один из важнейших показателей, влияющих на реальную производительность кластерной системы, это время, затраченное операционной системой и устройством на передачу одиночного пакета другому узлу кластера. Так как межузловой обмен данными происходит с помощью таких передач, то латентность можно охарактеризовать как время, потерянное процессом. Для задач с большим межузловым обменом большая латентность может дать катастрофическое падение производительности, в то же время для задач с малым межузловым обменом малая латентность не даст ничего в плане выигрыша производительности, но приведет к огромному увеличению бюджета проекта.
- Пропускная способность интерконнекта практически не сказывается на общей производительности системы. Существуют некоторые минимальные границы, но пропускная способность любого устройства, используемого сегодня в качестве интерконнекта, находится гораздо выше этих границ, т.е. пока что мы не встречали задачи, потребности которой в пропускной способности интерконнекта были бы сопоставимы с потребностями бенчмарка pingpong.

Надежность соединительных кабелей неявно влияет на реальную производительность системы в целом, так как может привести, и приводит, к серьезному увеличению латентности как на отдельном участке, так и во всей системе.

Выбранный интерконнект должен поддерживать запланированное количество узлов кластера (например, SCI ограничен 256 узлами), иначе будет невозможен выход на запланированную производительность или существенное падение реальной производительности в случае использования промежуточного интерконнекта.

Официальная производительность по пакету Linpack. Суперкомпьютеры, в том числе и кластеры, сравниваются между собой по результатам прогона пакета Linpack, а качество архитектуры и организации вычислительного процесса в кластере оценивается не только по отношению к максимально достижимой пиковой производительности, но в зависимости от установленного серверного оборудования и системного программного обеспечения.

Каждый прогон пакета Linpack на максимальных параметрах связан со значительными временными издержками (в пакете прогон одного набора исходных данных состоит в выполнении 18 тестов, каждый из которых может требовать как десятка минут для небольших матриц, с порядком до 40000, так и часов для матриц значительно большего размера). Благодаря размещению всех исходных данных в оперативной памяти суперкомпьютера и исключению свопинга данных из внешней памяти вполне приемлема технология предварительного расчета профиля производительности на матрицах небольшого размера для выбора локальных максимумов с последующим тщательным прогоном уже на этих максимумах. Грубое упрощение процедуры расчета сказывается, конечно, на полученных результатах, но общая картина представляется правильной, она во многом определяется размером NB пакета обмена, а также используемой технологией интерконнекта.

Относительно влияния технологии интерконнекта, можно констатировать стабильное значение измеренной производительности между тестами в рамках одного прогона для интерконнекта на InfiniBand’е и более чем 2 % разбросов для интерконнекта на основе SCI. Тем не менее, профили производительности в разных технологиях хорошо коррелируют друг с другом. Стартовый профиль производительности кластера СКИТ-1 на март 2005 г., относительно которого проводится сравнение, выглядит следующим образом (рис. 4, отметки в виде треугольника).

Аппаратура кластера была обновлена в октябре 2005 г. – увеличен объем оперативной памяти узла с 1 Гбайта до 2 Гбайт, добавлено еще 8 узлов и заменен интерконнект (вместо аппаратуры по технологии SCI и

соответствующих ей драйверов поставлена аппаратура и фирменное программное обеспечение по технологии InfiniBand). Интерес представляет не только изменение абсолютных показателей производительности, но и цифровое сопоставление оценок качества архитектуры кластера, как минимум, в виде максимально достигнутого коэффициента относительно пиковой производительности, а также сопоставление с результатами для кластеров аналогичной мощности и исходных аппаратных компонент по списку Top50[21].

Пиковая производительность кластера с 16 узлами на двух процессорах Xeon 2,67 ГГц в каждом составляет 170,24 Гфлопс (16 узлов х 2 процессора в узле х 2 конвейера в процессоре х 2,67 ГГц), что при максимально достигнутой производительности в тесте Linpack (112.6 Гфлопс) дает коэффициент 0,66. Это значение является наивысшим среди всех кластеров Top50 (редакция 2 от 05.04.2005 г.) с той же самой процессорной базой и оперативной памятью узла 1 Гбайт.

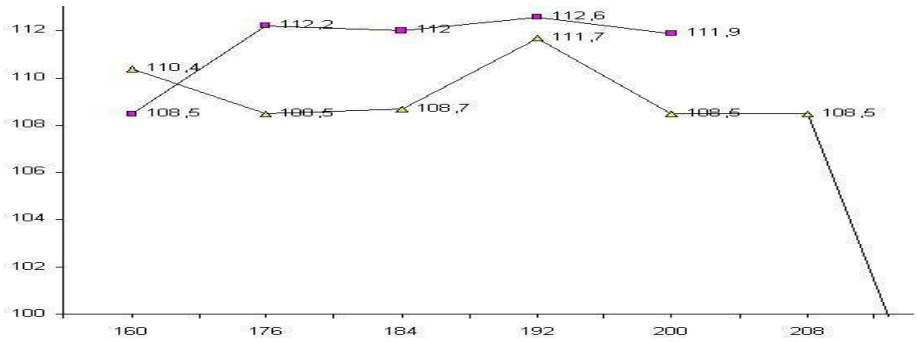

Рис. 4. Профиль производительности СКИТ-1

Увеличение размера оперативной памяти до 2 Гбайт увеличивает максимальное значение производительности 32-процессорного кластера до 126,4 Гфлопс и дает коэффициент 0,74. Это уже самое лучшее значение по Top50 (редакция 3 от 20.09.2005 г.) для кластеров с 32-битовыми процессорами и оперативной памятью узла в 1–2 Гбайта, лучшие цифры лишь для отдельных кластеров с оперативной памятью узла 4 Гбайта или 64-битовыми процессорами. С другой стороны, производительность на тех же исходных данных, что и в стартовом профиле, равна 117,7 Гфлопс и дает коэффициент 0,69, что с некоторыми допущениями напрямую характеризует качество нового интерконнекта кластера СКИТ-1.

Пиковая производительность обновленного кластера СКИТ-1 с 24 узлами на двух процессорах Xeon 2,66 ГГц в каждом составляет 255,6 Гфлопс (24 узла * 2 процессора в узле * 2 конвейера в процессоре * 2,66 ГГц) и при максимально достигнутой производительности в тесте Linpack 189,3 Гфлопс дает коэффициент 0,74 (эти значения переводят кластер СКИТ-1 с занимаемого 35 места в Top50, редакция 3 от 20.09.2005 г., на 25 место).

Многое из анализа производительности кластера СКИТ-1 можно отнести и к кластеру СКИТ-2, только цифры иные (пиковая производительность равна 358 Гфлопс, достигнутая максимальная производительность составляет 280 Гфлопс, коэффициент – 0,78).

6. Перспективы

В настоящее время производительность существующего кластерного комплекса СКИТ-1 и СКИТ-2 достаточна только для одновременного расчета нескольких задач, поэтому для удовлетворения имеющихся запросов со стороны Институтов НАН Украины, чтобы решение больших задач перестало быть узким местом, производительность комплекса следует поднять на порядок (до нескольких Терафлопс).

Оценочные характеристики кластерной системы, которая могла бы на ближайшие несколько лет отвечать насущным запросам упомянутых направлений в физике, биологии, технике и т.д., следующие:

1. количество современных двухядерных процессоров (с частотой 2,2 – 2,8 ГГц) – 300;
2. оперативная память (суммарная) — 1,0 Тбайт;
3. внешняя память на локальных магнитных дисках — 2.5 Тбайт;
4. система сохранение данных — 10 Тбайт;
5. суммарная пиковая производительность — 4,5 –5.5 Тфлопс.

Суммарная стоимость разработки такой суперЭВМ составляет 12–15 млн. гривен, включая затраты на разработку, закупку лицензионного программного обеспечения, холодильного оборудования, монтажно-строительные работы и т.п. Общая стоимость работ, включая создание прикладных программных пакетов в упомянутых интеллектуальных информационных технологиях, составляет 20–25 млн. гривен.

1. *Коваль В., Сергієнко І.* „СКІТ— український суперкомп'ютерний проект", Вісн. НАН України, 2005. - № 8. - С. 3─13.
2. *Koval V., Ryabchun S., Savyak V., Sergienko I., Yakuba A.* "SCIT — UKRAINIAN SUPERCOMPUTER PROJECT", International Conference KDS-2005 Proceedings, Varna, June 2005. - P. 98–104.
3. *Коваль В.Н., Рябчун С.Г., Сергиенко И.В., Якуба А.А.* «Суперкомпьютерный проект Института кибернетики им. В.М. Глушкова НАН Украины», ИИ, 2005. –№3. - С. 37–42.
4. www.lustre.org/

5. http://www.netlib.org/atlas/
6. http://www.netlib.org/blacs/
7. http://www.netlib.org/scalapack/
8. http://www.intel.com/cd/software/products/asmo-na/eng/perflib/mkl/index.htm
9. http://www.gromacs.org/
10. http://www.wien2k.at/
11. www.msg.ameslab.gov/GAMESS/GAMESS.html
12. http://www.scali.com/
13. https://docs.mellanox.com/dm/ibgold/ReadMe.html
14. http://www.lam-mpi.org/
15. http://www.open-mpi.org/
16. www.t-platforms.ru/english/about/dnd.html
17. www.intel.com/design/servers/ipmi/spec.htm
18. www.redhat.com/software/rha/gfs/
19. www.openafs.org/
20. oss.oracle.com/projects/ocfs2/
21. www.supercomputers.ru/