УДК 004.724.4(045)

Аленин О.И.

АНАЛИЗ ВЛИЯНИЯ ПАРАМЕТРОВ СЕТИ ПЕРЕДАЧИ ДАННЫХ НА ПРОИЗВОДИТЕЛЬНОСТЬ ВИРТУАЛЬНОГО РАСПРЕДЕЛЕННОГО КЛАСТЕРА

Национальный технический университет Украины "КПИ"

Рассмотрены вопросы, возникающие при построении виртуальных кластерных систем с территориально удаленными узлами. На основании проведенных экспериментов сделан анализ, показывающий влияние параметров глобальной сети (такие как задержка, пропускная способность) на производительность таких кластеров. Даны рекомендации относительно целесообразности использования дополнительных удаленных узлов для решения задачи

Введение

Для решения научно-исследовательских задач и проведения инженерных расчетов всегда требовались большие вычислительные мощности. В последнее время, наибольшую часть среди высокопроизводительных вычислительных систем составляют системы с кластерной архитектурой [1], это объясняется, прежде всего, их относительно невысокой стоимостью при высокой производительности в сравнении с системами других типов.

Для достижения еще большей производительности предпринимаются попытки объединить вычислительные ресурсы нескольких систем в один вычислительный ресурс. Так возник подход распределенных вычислений и *grid*.

Однако у существующих на сегодняшний день реализаций *grid*-систем есть ряд недостатков. Во-первых, они сложны в установке и настройке (требуют установки специальной операционной системы и набора системных программ). Во-вторых, не всякое прикладное программное обеспечение может использоваться в *grid*-среде.

Большую часть параллельных прикладных программ для научных и инженерных расчетов предполагалось использовать на кластерных системах. В третьих, существующие реализации *grid* не предоставляют возможности использовать для решения одной задачи ресурсы нескольких кластеров, участвующих в *grid*, одновременно.

Обзор существующих решений

Существует несколько подходов к решению первых двух проблем, и все они опираются на использование виртуализации.

В работе [2] сказано, что эти проблемы могут быть решены путем создания кластеров виртуальных машин, с соответствующим набором программного обеспечения, необходимого для запуска приложений. Описано конфигурирование, выполнение и оценка виртуальных сред в контексте виртуального кластера для *Open Science Grid*.

Оценка производительности показывает, что такие среды действительно могут быть использованы для *grid*-приложений.

В работе также показано, что при использовании виртуальных машин для запуска приложений потери производительности из-за использования виртуализации не превышают 5%.

В работе [3] сказано, что виртуальные машины – удобное средство контролируемого совместного использования ресурсов, позволяющее создавать виртуальные ресурсы с заданными параметрами, необходимым предустановленным программным обеспечением и набором аппаратных ресурсов.

Описана модель предоставления ресурсов для *grid*-среды, которая позволяет описывать такие виртуальные ресурсы в составе *grid*-инфраструктуры.

также, что учитывая время на создание и развертывание виртуальной среды в планировщике, а не оставляя этот процесс пользователю, можно достичь значительно большей эффективности использования ресурсов и более точного соответствия требованиям ко времени выполнения заданий.

Учет в планировщике времени на передачу и развертывание образов виртуальных машин имеет два преимущества: во-первых, передачу образа можно запланировать и выполнить заранее, во-вторых, можно кешировать часто используемые образы виртуальных машин.

В работе [4] предложен прототип системы управления виртуальным кластером, а также использование технологий *iSCSI* для виртуализации дисковых ресурсов и *VLAN* для сетевых ресурсов.

Для виртуализации узлов используется *VMWare*. Система позволяет получать полностью сконфигурированные виртуальные кластеры с предустановленным программным обеспечением.

Выделяется такие типы узлов:

– управляющий – на нем работает программное обеспечение, управляющее всеми физическими узлами и виртуальными кластерами;

– шлюз – на нем работают управляющие узлы виртуальных кластеров;

– вычислительный узел – на нем работают узлы виртуальных кластеров;

– хранилище – узел с большим дисковым пространством, предоставляющий доступ к нему по *iSCSI*. Гибкое управление дисковым пространством осуществляется с помощью *LVM*.

В работе [5] сказано, что в настоящее время всё большее распространение получает использование виртуальных машин для научных приложений, и появилась необходимость включить возможность предоставления ресурсов в виде виртуальных машин в состав стандартных систем управления ресурсами. Для решения этой задачи разработан способ управления виртуальными машинами, использующий многоуровневое планирование и позволяющий включить в такие планировщики как *PBS* эту возможность. В работе показано, что при условии наличия образов виртуальных машин на физических узлах развертывание виртуального кластера занимает относительно небольшое время (12 секунд для кластера из 16 узлов, включая время загрузки виртуальных узлов).

В работе [6] предложена структура системы управления ресурсами, в которой применяется подход к предоставлению ресурсов с помощью виртуальных машин.

При таком подходе появляются возможности приостанавливать и возобновлять выполнение задач, переносить задачи вместе с виртуальными машинами на другие физические узлы, а также предоставлять вычислительные ресурсы (виртуальные машины) с предустановленным программным обеспечением, необходимым для выполнения данной задачи.

В статье также показано, что подход с использованием виртуальных машин позволяет повысить достичь большей общей производительности системы.

Проблема эффективного объединения вычислительных ресурсов территориально удаленных кластеров для решения одной задачи не решена до сих пор. В частности, раньше это было связано с недостаточной пропускной способностью каналов передачи данных в глобальных сетях.

С появлением новых технологий передачи данных и увеличению пропускной способности каналов связи, появляется возможность применять новые подходы для решения этой проблемы.

Постановка задачи

Для эффективного использования всех доступных ресурсов нескольких кластерных систем, в общем случае соединенных между собой через глобальную сеть, а также для предоставления возможности выполнения параллельной прикладной программы, рассчитанной на работу на кластерной системе, на нескольких таких системах одновременно, предлагается создать виртуальный ,

использующий физические ресурсы этих систем. Для пользователей и прикладных программ такая система будет представляться как кластер большего размера и большей производительности.

Причем, необходимо учитывать требования конкретного прикладного программного обеспечения как к аппаратным ресурсам, таким как количество узлов, процессоров, объем оперативной памяти, дисковой памяти, так и к программной среде – операционной системе, библиотекам.

В общем случае, физические кластеры одновременно могут участвовать в нескольких виртуальных кластерах.

Есть несколько подзадач, которые надо решить в процессе построения такого виртуального кластера:

– создание на основе неоднородных физических однородных виртуальных ресурсов;

– создание системы управления виртуальными и физическими ресурсами;

– объединение узлов каждого виртуального кластера в одну подсеть.

В зависимости от операционной системы на физическом кластере и операционной системы, требующейся для приложения, необходимо выбрать тип виртуализации.

Кроме того, от выбранного типа виртуализации зависят накладные расходы на виртуализацию, и, как следствие, потеря производительности по сравнению с физическим ресурсом. Сравнение типов виртуализации, их достоинства и недостатки описаны в [7].

Для разработки системы управления ресурсами такого виртуального кластера, построенного на базе территориально удаленных физических кластеров, а также алгоритмов ее работы, необходимо предварительно изучить, как особенности глобальной сети влияют на эффективность работы такого кластера.

Целью данной работы является построение модели виртуального кластера, и анализ влияния таких параметров, как задержка и пропускная способность канала на производительность кластера при решении эталонной задачи.

Решение задачи

Для проведения экспериментов было использовано несколько компьютеров с *ОС Linux* в качестве узлов кластера, и программный маршрутизатор-компьютер в *ОС FreeBSD* и системой *dummynet* в качестве эмулятора глобальной сети [8]. Физические сетевые интерфейсы на узлах были настроены таким образом, что взаимодействовать друг с другом они могли только через маршрутизатор.

На узлах были настроены виртуальные интерфейсы, и построена виртуальная подсеть. На маршрутизаторе задавались необходимые значения задержки и ограничения пропускной способности каналов.

В качестве эталонной задачи использовался тест *linpack* [9], который первоначально являлся дополнением к одноименной библиотеке численных методов, содержащей набор процедур для решения систем линейных алгебраических уравнений и предназначался для оценки времени решения той или иной системы с помощью этой библиотеки.

Linpack является классическим примером теста-ядра (причем, поскольку к решению тех или иных СЛАУ сводятся очень многие реальные расчетные задачи – измеренные им характеристики являются в высокой степени репрезентативными).

Тест состоит в решении системы линейных арифметических уравнений вида $Ax=f$ методом LU-факторизации с выбором ведущего элемента столбца, где A – плотно заполненная матрица размерности N (первоначальный вариант *Linpack* решал задачу размерности 100).

Производительность в тесте *Linpack* измеряется в количестве производимых операций с плавающей запятой в секунду. Единицей измерения является 1 Флопс.

С течением времени и увеличении вычислительной мощности компьютеров размерность теста *Linpack* была увеличена до 1000. Однако с появлением все -

лее мощных вычислительных систем и эта размерность стала чересчур малой, более того, для тестирования кластерных систем была создана отдельная версия теста *HPL* [10] в которой размерность матрицы (и некоторые другие параметры) не являются фиксированными, а задаются пользователем теста.

При увеличении размерности матрицы решаемой задачи, растет степень параллелизма, что может привести к увеличению производительности. Другим важным параметром, влияющим на производительность, является размер блока, с которым матрица распределяется между узлами кластерной системы.

Данный тест является общепринятым стандартом для оценки производительности кластерных систем.

На графиках производительность показана по отношению к производительности кластера (она принята за единицу), части которого соединяются без эмулятора глобальной сети.

На рис. 1 показана зависимость производительности системы от задержки в сети при фиксированном значении размерности задачи и размере блока данных. Значение задержки изменяется от 0 до 200 мс. Как видно из графика, производительность сильно падает при увеличении задержки, и при задержке около 25 мс производительность падает на 25%. При задержке 50 мс производительность уменьшается в 2 раза.

На рис. 2 показана зависимость производительности от пропускной способности каналов.

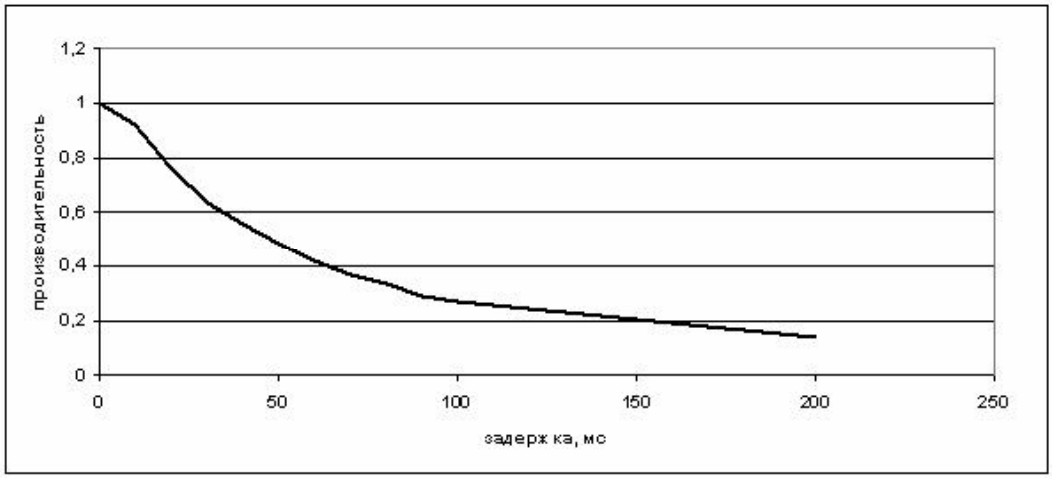

Рис. 1. Зависимость производительности от задержки в сети

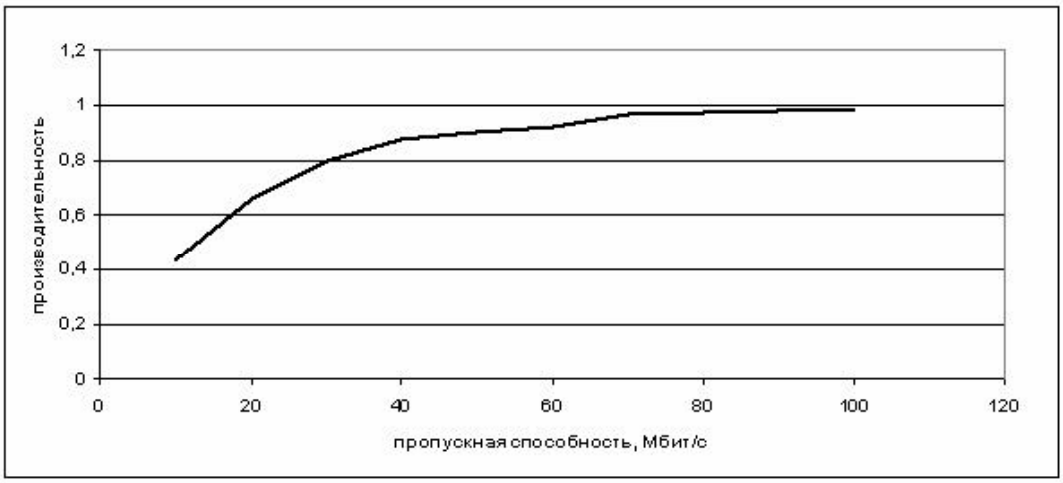

Рис. 2. Зависимость производительности от пропускной способности

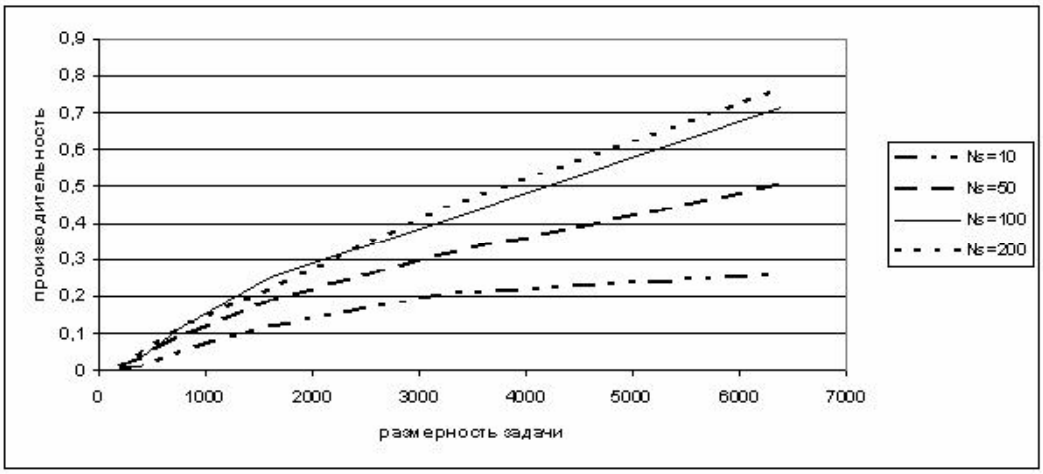

Рис. 3. Зависимость производительности размерности задачи

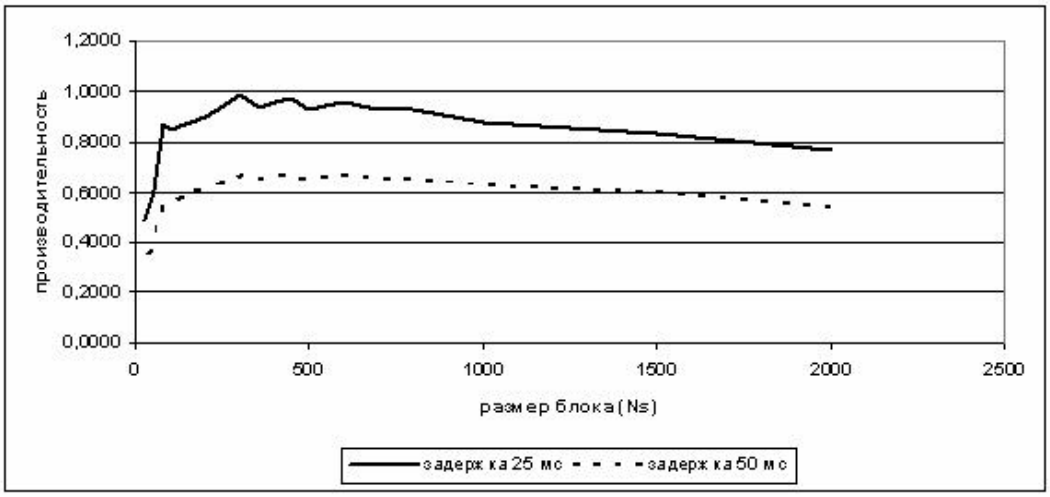

Рис. 4. Зависимость производительности размера блока данных

Значение пропускной способности каналов меняется от 10Мбит/с до 100Мбит/с. После значения около 60Мбит/с дальнейшее увеличение пропускной способности практически не влияет на производительность.

На рис. 3 показана зависимость производительности при изменении размерности задачи.

Здесь зафиксирована пропускная способность каналов (100Мбит/с) и задержка (25 мс). Несколько графиков показывают эти зависимости при разных значениях блока данных.

Из графика видно, что при маленьком блоке данных, и, следовательно, частых обменах данными между процессами, работающими на разных узлах, время решения задачи существенно больше, чем при оптимальном размере блока.

Для задач большой размерности (N=6400) при размере блока Ns=10 время в 2.5 раза больше, чем при Ns=100.

С ростом размера блока выигрыш в скорости решения задачи становится меньше.

Из графика видно, что имеет значение не столько собственно размер блока, сколько отношение N/Ns.

Для более точного определения оптимального соотношения рассмотрим зависимость производительности и времени решения задачи при изменении значения размера блока (рис. 4), зафиксировав значение размерности задачи (Ns), задержки в сети и пропускной способности.

Показаны графики для пропускной способности 100Мбит/с и двух значений задержки – 25 и 50 .

Из графика видно, что с увеличением размера блока производительность сначала резко увеличивается, доходит до максимума, а потом начинает медленно падать.

На промежутке значений от *Ns*=100 до *Ns*=1000 (от *N/Ns*=1/60 до *N/Ns*=1/6) производительность меняется незначительно.

Выводы

1. Каналы передачи данных с пропускной способностью 100Мбит/с и задержкой до 25 мс являются допустимыми для объединения территориально удаленных узлов кластера. Задержка сильно влияет на производительность системы, и при ее значении выше 50 мс производительность падает в 2 раза. При этом выигрыш от использования удаленных узлов отсутствует.

2. Время решения задачи зависит также от размера блока. Маленький размер блока приводит к частым обращениям для передачи данных, и, даже если общий объем передаваемых данных остается неизменным, время решения задачи существенно возрастает.

3. В целом, с использованием современных каналов передачи данных возможно объединять территориально удаленные узлы в виртуальный кластер для запуска *MPI*-программ.

Список литературы

1. TOP500 list of supercomputers [Online source] – 2008. – Link: http://www.top500.org/lists/2008/11.

2. *Foster I.* Virtual Clusters for Grid Communities / I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, X. Zhang // CCGRID 2006, 16-19 May 2006.: thesis rep. – Singapore, 2006. – P. 513–520.

3. *Sotomayor B.* Overhead Matters: A Model for Virtual Resource Management / B. Sotomayor, K. Keahey, I. Foster // VTDC 2006, 17 Nov 2006.: thesis rep. – Tampa (USA), 2006. – P. 1–5.

4. *Nakada H.* The design and implementation of a virtual cluster management system / H. Nakada, T. Yokoi, T. Ebara, Y. Tanimura, H. Ogawa, S. Sekiguchi // IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management (EVGM2007), 2007.

5. *Freeman T.* Flying Low: Simple Leases with Workspace Pilot / Freeman, T., K. Keahey // Euro-Par 2008, 29-30 October 2008.: thesis rep. – Las Palmas de Gran Canaria, Canary Island (Spain), 2008.

6. *Sotomayor B.* Combining Batch Execution and Leasing Using Virtual Machines / B. Sotomayor, K. Keahey, I. Foster // 17th international symposium on High performance distributed computing HPDC2008. June 2008.: thesis rep. – Boston (USA), 2008, – P. 87–96.

7. *Jones T.* An overview of virtualization methods, architectures, and implementations [Online source] /T.Jones // IBM – 2006. – Link: http://www.ibm.com/developerworks/linux/library/l-linuxvirt.

8. Dummynet – traffic shaper, bandwidth manager and delay emulator [Online source] / FreeBSD Kernel Interfaces Manual. – 2002. – Link: http://www.freebsd.org/docs.html.

9. *Свистунов А.Н.* Оценка производительности кластерных систем. Учебный курс / А.Н. Свистунов // Нижегородский государственный институт им. Лобачевского. – Нижний . – 2007. – Link: www.software.unn.ru/ccam/multicore/materials/cluster/performance_test.pdf

10. HPL – A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers [Online source] / A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary // Netlib Repository. – 2008. – Link: http://www.netlib.org/benchmark/hpl/.