УДК 004.272.43.003.13

Е.А. Башков [1], д.т.н., проф.,
В.П. Иващенко [2], д.т.н., проф.,
Г.Г. Швачич [2], к.т. н., доц.
[1]Донецкий национальный технический университет, г. Донецк, Украина
bashkov@pmi.dgtu.donetsk.ua,
[2]Национальная металлургическая академия Украины, г. Днепропетровск, Украина
sgg1@ukr.net

Реализация режима агрегации каналов сетевого интерфейса в модульных многопроцессорных вычислительных системах

Статья посвящена проблеме повышения эффективности многопроцессорных кластерных систем за счет реорганизации архитектуры ее сетевого интерфейса. Предложенный подход позволил существенно уменьшить время граничного обмена данных между вычислительными узлами кластерной системы, что дает возможность не только повысить эффективность распараллеливания, но и существенно уменьшить время вычислений.

Ключевые слова: многопроцессорная вычислительная система, сетевой интерфейс, эффективность распараллеливания, ускорение вычислений, коммутатор.

Введение

В настоящее время существует много различных вариантов построения кластерных вычислительных систем. В данной работе рассматриваются так называемые "блейд" серверные решения многопроцессорных систем [1,2]. Однако одно из основных различий в их конструировании лежит в области используемой сетевой технологии, выбор которой определяется, прежде всего, классом решаемых задач.

Например, в задачах металлургии при математическом моделировании скоростных режимов термической обработки длинномерных изделий одна из основных проблем может быть сформулирована так: имеем разностную сетку размерности *M*, время вычисления задачи, которая решается с использованием однопроцессорной системы, определяется величиной *t*. Этот параметр является критичным. Необходимо существенно уменьшить время вычислений, сохраняя значение *M*. В этой связи, вопросам быстродействия, эффективности и производительности уделяется основное внимание при конструировании кластерных систем.

Итак, рассматривается задача уменьшения времени расчетов путем увеличения числа узлов кластерной системы. Такой подход ориентирован, например, на разработку новых технологических процессов (когда время вычислений является собой критическую величину) [3, 4, 5]. Кроме того, аналогичные задачи часто приходится решать в экономике, медицине, военной технике и др.

Таким образом, тема конструирования кластерных многопроцессорных систем на сегодняшний день является актуальной, интересной и переживает этап своего бурного развития. Ясно и другое, что при помощи высокопроизводительных кластеров найден эффективный способ решения широкого класса актуальных задач.

По нашему мнению, новый качественный этап развития многопроцессорных кластерных систем лежит в области использования новых современных сетевых технологий. При этом эффективность распараллеливания вычислений зависит от многих факторов, однако одним из определяющих является выбор и организация сетевого интерфейса. Это объясняется следующим образом. Сеть кластерной вычислительной системы принципиально отличается от сети рабочих станций, хотя для построения кластера необходимы обычные сетевые карты и хабы/коммутаторы, которые применяются при организации сети рабочих станций. Однако в случае кластерной вычислительной системы имеется одна принципиальная особенность. *Сеть кластера, в первую очередь, предназначена не для связи компьютеров, а для связи вычислительных процессов*. В этой связи, чем выше будет пропускная способность вычислительной сети кластера, тем быстрее будут считаться пользовательские параллельные задачи, выполняемые на кластере. Таким образом, технические характеристики вычислительной сети при-

обретают первостепенное значение для многопроцессорных кластерных систем.

В настоящее время проблема реализации режима агрегации каналов сетевого интерфейса для модульных многопроцессорных кластерных систем не получила должного развития. Кроме того, практически отсутствуют работы, посвященные исследованию влияния архитектуры сети кластерной системы на эффективность распараллеливания. В этой связи, рассматриваемые в данной работе исследования являются актуальными, и, несомненно, вызовут интерес у соответствующих специалистов.

Проблема и задачи исследований

Анализ режимов функционирования сетевого интерфейса многопроцессорных систем позволяет сформулировать следующую проб-лему: *каким образом за счет конструктивных особенностей архитектуры сетевого интерфейса многопроцессорных кластерных систем можно повысить оценки ее эффективности и быстродействия?*

Такая проблема может быть решена следующим образом. Для увеличения пропускной способности сети кластера рекомендуется применять процедуру "связывания каналов" или технологию *channel bonding*. Технология связывания каналов (*channel bonding*) позволяет объединять несколько сетевых адаптеров в один скоростной канал. При этом обмен данных между вычислительными узлами кластера выносится в отдельную сеть, которая работает на канальном (втором) уровне с использованием технологии *channel bonding*. Такой подход направлен на увеличение скорости обмена данными между узлами кластера, и снижение загрузки канала, который соединяет узлы кластера.

Кроме того, введение дополнительных управляемых коммутаторов, которые работают параллельно, позволяет через терминал или *WEB*-интерфейс изменять конфигурацию сети, повышать ее пропускную способность. Такая архитектура сети обеспечит высокоскоростной доступ к памяти узлов кластерной системы. Вообще отметим, что реализация реконфигурируемой сети позволяет повысить эффективность кластерной системы, *адаптируя структуру ее сети для решения каждого конкретного типа задач.*

Итак, предлагаемая сетевая архитектура многопроцессорной кластерной системы должна позволять, во-первых, повысить быстродействие вычислений во время решения сильносвязанных задач и, во-вторых, обеспечить высокоскоростной доступ к памяти узлов кластера, снижая загрузку канала, который проходит между узлами вычислительной системы.

В результате проведенных исследований необходимо решить следующие задачи:

− выявить пути повышения эффективности многопроцессорной кластерной системы за счет реорганизации архитектуры ее сетевого интерфейса;

− вывести аналитические соотношения для определения оптимального числа узлов кластерной системы для различных режимов ее функционирования;

− для удобства определения оценок эффективности кластерной вычислительной многопроцессорной системы вывести основные аналитические соотношения через параметры исследуемой системы.

Особенности сопряжения вычислительных узлов с сетевым интерфейсом для режима агрегации каналов сетевого интерфейса многопроцессорной системы

Применяемая технология связывания каналов сетевого интерфейса многопроцессорной кластерной системы позволяет объединить узлы кластера в сеть таким образом, чтобы каждый узел многопроцессорной системы подсоединялся к коммутатору более чем одним каналом. Для реализации такого подхода необходимо оснастить узлы кластера либо несколькими сетевыми платами, либо многопортовыми платами. Связывание каналов аналогично режиму транкинга при соединении коммутаторов, который используется для увеличения скорости передачи данных между двумя или несколькими коммутаторами. Применение процедуры связывания каналов под управлением ОС *Linux* позволяет организовать равномерное распределение нагрузки (приема/передачи данных) между соответ-ствующими каналами многопроцессорной системы и увеличить скорость обмена данных между ее узлами.

Вообще заметим, что технология *агрегации каналов* может порождать некоторые проблемы, связанные с выбором коммутаторов и их настройкой. Например, коммутатор должен поддерживать режим связывания каналов, иначе могут иметь место всевозможные ошибки при построении коммутатором таблиц маршру-тизации пакетов или таблиц *MAC*-адресов. Такие коммутаторы должны поддерживать для своих портов функции *Link Aggregation* или соответствовать стандарту *IEEE 802.3ad*.

Другим вариантом реализации технологии агрегации каналов сетевого интерфейса может быть выбор коммутатора с возможностью поддержки режима виртуальных локальных сетей (*VLAN*). Применение *VLAN* призвано помочь избежать "дублирования" во внутренних таблицах коммутаторов *MAC*-адресов многопортовых сетевых плат. Впрочем, практика показывает, что и поддержка режима *VLAN* не всегда помогает эффективно разделить каналы.

Вообще заметим, что возможен и иной прием формирования многоканального сетевого интерфейса. Так, в [6] показано, что можно отказаться от применения специализированного сетевого оборудования, поддерживающего связывание каналов. Сетевые каналы при этом можно организовать при помощи двойного (тройного и т.д.) набора обычных хабов или свитчей. Здесь непересекающиеся сетевые сегменты организуются таким образом, чтобы каждый новый сетевой канал образовывал свою собственную сеть, физически не связанную с сетями других каналов.

Несмотря на широкий выбор методов формирования многоканального режима функционирования сетевого интерфейса, в многопроцессорных вычислительных системах приведем некоторые принципиальные особенности, которые следует иметь в виду при конструировании режима агрегации сетевого интерфейса. Так, все процессоры в подсети должны быть объединены одинаковым способом. Объединение каналов требует, как минимум, двух физических подсетей, которых, тем не менее, может быть и больше. Заметим, что в версиях ядра ОС `2.4.x` технология связывания каналов сетевого интерфейса является стандартной опцией ядра. Сетевые карты настраиваются, как обычно, за исключением того, что команду *ifconfig* необходимо применять для активации первой сетевой карты в связке. Утилита *ifenslave* используется для активации оставшихся сетевых карт связанного соединения. Сети с объединенными каналами могут соединяться с обычными сетями посредством маршрутизатора или моста, поддерживающего технологию *Channel Bonding*.

Схема организации сетевого интерфейса по технологии *channel bonding* для рассматриваемой многопроцессорной системы приведена на рис. 1.

На первом этапе освещения сетевого интерфейса рассмотрим особенности конфигурации сети 1. Коммутатор *SW*1 образует сеть управления, загрузки и диагностики кластера. Так, интегрированный сетевой интерфейс *PM*001.*i*01 узла *MNode*001 с функцией сетевой загрузки подсоединен входом/выходом к порту 01 управляемого коммутатора *SW*1. Интегрированный сетевой интерфейс *PN*001.*i*01 узла *PNode*001 с функцией сетевой загрузки подсоединен входом/выходом к порту 02 управляемого коммутатора *SW*1. Интегрированный сетевой интерфейс *PN*002.*i*01 узла *PNode*002 с функцией сетевой загрузки подсоединен входом/выходом к порту 03 управляемого коммутатора *SW*1. Интегрированный сетевой интерфейс *PN*003.*i*01 узла *PNode*003 с функцией сетевой загрузки подсоединен входом/выходом к порту 04 управляемого коммутатора *SW*1. Интегрированный сетевой интерфейс *PN*004.*i*01 узла *PNode*004 с функцией сетевой загрузки подсоединен входом/выходом к порту 05 управляемого коммутатора *SW*1. Интегрированный сетевой интерфейс *PN*005.*i*01 узла *PNode*005 с функцией сетевой загрузки подсоединен входом/выходом к порту 06 управляемого коммутатора *SW*1. Мастер – узел *MNode*001 соединен входом/выходом двунаправленным внешним сетевым интерфейсом *Gi*_001,1 к порту 12 управляемого коммутатора *SW*1. Мастер – узел *MNode*001 соединен входом/выходом двунаправленным внешним сетевым интерфейсом *Gi*_001,2 с сетью *Ethernet*. Порт 11 управляемого коммутатора *SW*1 соединен с портом 12 сетевого интерфейса управляемого коммутатора *SW*2, а порт 10 управляемого коммутатора *SW*1 соединен с портом 12 управляемого коммутатора *SW*3 и образуют подсеть *vp*123 для *Web* конфи-гурирования и диагностики коммутаторов.

Особенности организации сетевого интерфейса для режима агрегации каналов (рис. 1) состоят в том, что в сети обмена данными многопроцессорной вычислительной системы сконфигурированы две симметрично работающие вычислительные сети (сеть 2 и сеть 3) на основе двух коммутационных матриц (коммутаторы *SW* 2 и *SW* 3). Архитектура вычислительной сети кластера для реализации режима *channel bonding* организована следующим образом.

Slave – узел *PNode*001 соединен входом/выходом двухпортовым двунаправлен-ным внешним сетевым интерфейсом *Gi*_001,1 портом 1 с портом 01 управляемого коммутатора *SW*2, а портом 2 с портом 01 управляемого коммутатора *SW*3. Кроме того, дополнительно такой узел соединен двухпортовым двунаправленным сетевым интерфейсом *Gi*_001,2 портом 1 с портом 02 управляемого коммутатора *SW*2, а портом 2 с портом 02 управляемого коммутатора *SW*3.

Slave – узел *PNode*002 соединен входом/выходом двухпортовым двунаправлен-ным внешним сетевым интерфейсом *Gi*_002,1 портом 1 с портом 03 управляемого ком-мутатора *SW*2, а портом 2 с портом 03 управляемого коммутатора *SW*3. Кроме того, дополнительно такой узел соединен двухпортовым двунаправленным сетевым интерфейсом *Gi*_002,2 портом 1 с портом 04 управляемого коммутатора *SW*2, а портом 2 с портом 04 управляемого коммутатора *SW*3.

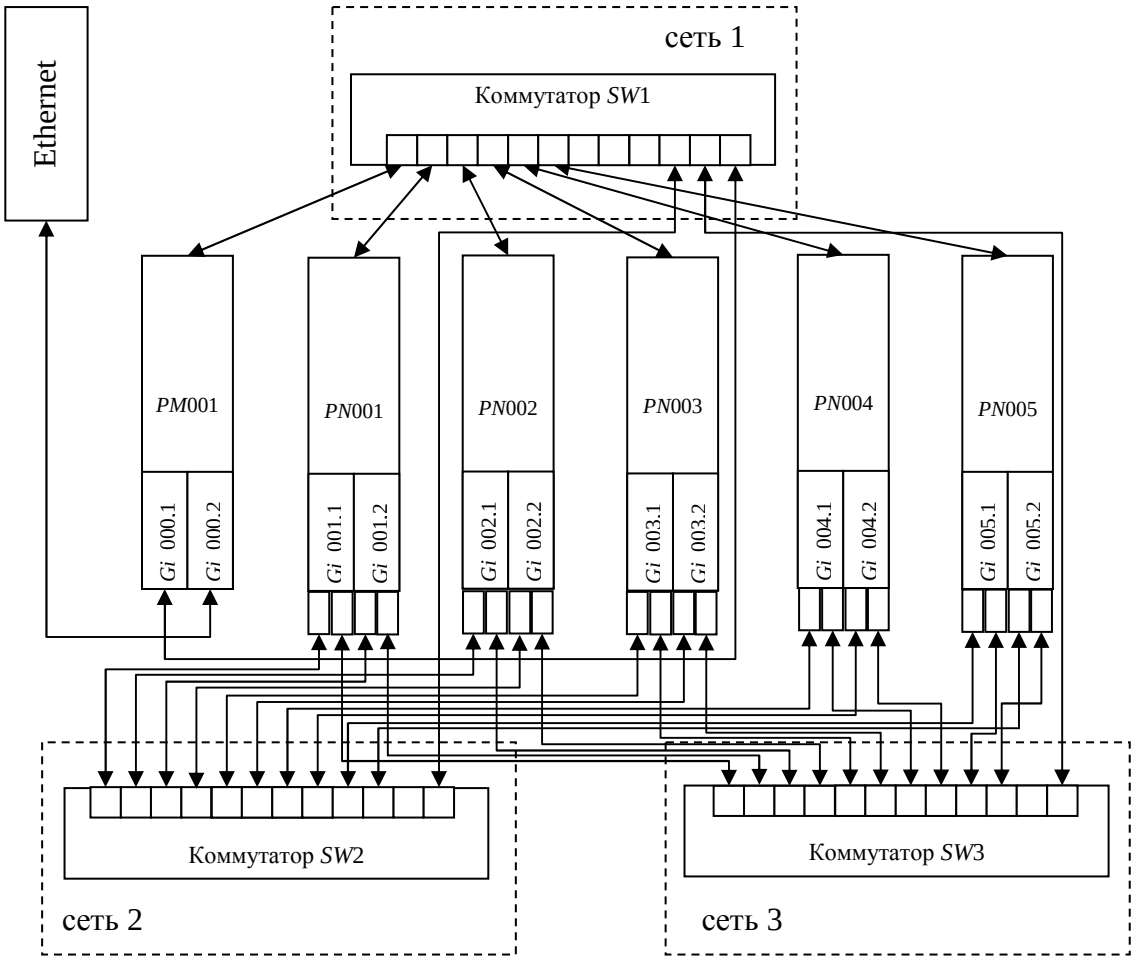

Рисунок 1 – Схема организации сетевого интерфейса по технологии channel bonding

Slave – узел *PNode*003 соединен входом/выходом двухпортовым двунаправлен-ным внешним сетевым интерфейсом *Gi*_003,1 портом 1 с портом 05 управляемого коммутатора *SW*2, а портом 2 с портом 05 управляемого коммутатора *SW*3. Кроме того, дополнительно такой узел соединен двух- портовым двунаправленным сетевым интерфейсом *Gi*_003,2 портом 1 с портом 06 управляемого коммутатора *SW*2, а портом 2 с портом 06 управляемого коммутатора *SW*3.

Slave – узел *PNode*004 соединен входом/выходом двухпортовым двунаправлен-ным внешним сетевым интерфейсом *Gi*_004,1 портом 1 с портом 07 управляемого коммутатора *SW*2, а портом 2 с портом 07 управляемого коммутатора *SW*3. Кроме того, дополнительно такой узел соединен двух- портовым двунаправленным сетевым интерфейсом *Gi*_004,2 портом 1 с портом 08 управляемого коммутатора *SW*2, а портом 2 с портом 08 управляемого коммутатора *SW*3.

Slave – узел *PNode*005 соединен входом/выходом двух портовым двунаправленным внешним сетевым интерфейсом *Gi*_005,1 портом 1 с портом 09 управляемого коммутатора *SW*2, а портом 2 с портом 09 управляемого коммутатора *SW*3. Кроме того, дополнительно такой узел соединен двухпортовым двунаправленным сетевым интерфейсом *Gi*_005,2 портом 1 с портом 10 управляемого коммутатора *SW*2, а портом 2 с портом 10 управляемого коммутатора *SW*3.

Приведенная схема организации сетевого интерфейса включает на каждом вычислительном узле по две однотипные двух- портовые сетевые карты и два однотипных коммутатора. Для конфигурации приведенных сетевых интерфейсов выполняются основные операции по настройке режима *Link Aggregation*.

Для решения широкого круга прикладных задач граничный обмен данных осуществляется между соседними узлами. В таком случае связь между узлами кластера организуется по топологии кольцо (рис. 2), т.е. узел *PN*001 обменивается данными с *PN*002, узел *PN*002 с *PN*003, узел *PN*003 с *PN*004, узел *PN*004 с *PN*005, узел *PN*005 с *PN*001.

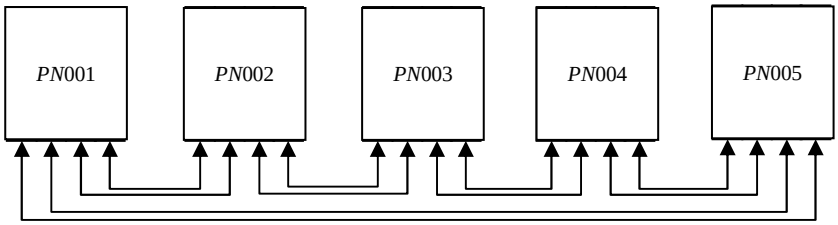

Рисунок 2 – Структура вычислительной сети кластера для реализации граничного обмена в режиме агрегации каналов

Благодаря такому подходу, появляется возможность, с одной стороны, организовать равномерное распределение нагрузки (приема/передачи данных) между соответствующими узлами кластерной системы, а с другой – увеличить скорость обмена данными между узлами кластерной системы. Очевидно, что, чем выше будет пропускная способность сети, тем быстрее будут решаться параллельные задачи, обрабатываемые при помощи модульной кластерной системы.

Особенности организации и настройки многоканального сетевого интерфейса многопроцессорной системы

Приведем некоторые важные особенности организация сетевого интерфейса в соответствии с технологией *channel bonding*. В этой связи рассмотрим сам принцип связывания в параллель нескольких *Ethernet*-адаптеров. Допустим, есть два адаптера *Ethernet*: *eth*0 и *eth*1. Их необходимо объединить в псевдо*Ethernet*-адаптер *eth*3. При этом система распознает эти агрегированные адаптеры как один. Все агрегированные адаптеры настраиваются на один *MAC*-адрес, поэтому удаленные серверы обращаются с ними как с одним адаптером. Псевдоадаптер *eth*3 можно настроить на один *IP* адрес, как любой *Ethernet*-адаптер. Из-за этого программы обращаются к нему как к самому обычному адаптеру, скорость которого в два раза выше. Протоколы агрегирования сетевого интерфейса определяют, какие порты используются для исходящего трафика или какой конкретный порт принимает входящий трафик. Состояние интерфейса используется для проверки линка, является он активным или нет.

В связи с приведенными особенностями связывания каналов, укажем требования, предъявляемые к аппаратным и программным средствам многопроцессорной кластерной системы. Так, все лезвия многопроцессорной системы должны иметь одинаковый набор *bonded networks*, т.е. нельзя на одном лезвии использовать сетевую карту типа 2*x*100*BaseTx*, а на другом – 10*Base* и 100*BaseTx*. Режим работы сетевых карт тоже должен быть единообразным. Другими словами, недопустим вариант, когда одна сетевая карта функционирует в режиме *full duplex*, а другая - в полудуплексном режиме. Технология *channel bonding* требует наличия, как минимум, двух физических подсетей. Но, при необходимости, связанный канал можно построить на основе трех или более сетевых карт.

Для связывания сетевых карт в один канал (одну виртуальную карту) необходимо либо скомпилировать ядро ОС с поддержкой режима *channel bonding*, либо загрузить в ОС модуль ядра *bonding.o*.

Заметим, что в ОС *Linux*, начиная с ядер 2.4.*x*, технология *channel bonding* является стандартной включаемой опцией. Например, в дистрибутиве *Alt Linux Master* 2.2 технология *channel bonding* поставляется в виде загружаемого модуля ядра.

Для конфигурации связанного канала требуется стандартная команда *ifconfig* и, возможно, дополнительная команда *ifenslave*. Это объясняется тем, что программа *ifenslave* позволяет копировать установки первого интерфейса на все остальные дополнительные интерфейсы.

Для рассматриваемой многопроцессорной системы был реализован режим формирования двух подсетей. В этой связи, процесс настройки технологии *channel bonding* в настоящей статье будет приведен на примере использования двух сетевых карт. Сетевой интерфейс для первой карты должен быть заранее сконфигурирован и полностью работоспособен. Для добавления в систему второй сетевой карты и объединения ее с первой в связанный канал требуется выполнить некоторые операции.

Предварительно останавливаются сетевые интерфейсы в многопроцессорной системе при помощи команды

```
/etc/rc.d/init.d/network stop
```

После этого переходят собственно к конфигурации связанного канала. На первом этапе необходимо изменить файл */etc/modules.conf*, добавив в него следующую строку:

```
alias bond0 bonding
```

Такое добавление сообщает системе о том, что необходимо загрузить модуль *bonding.o*, который определяется также по алиасу *bond*0. Чтобы не перезагружать систему, вручную загружается модуль

```
modprobe bonding
```

Далее переходят в каталог */etc/sysconfig/network-scripts* и переименовывают файл описания первого интерфейса *ifcfg-eth0* в *ifcfg-bond0*:

```
cp ifcfg-eth0 ifcfg-bond0
```

Сформированный файл *ifcfg-bond0* необходимо отредактировать так, чтобы он принял следующий вид:

```
DEVICE=bond0
IPADDR=192.168.1.*
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
```

Конечно, пользователь должен указать здесь свои собственные *IP*-адрес, маску, адрес сети и *broadcast*. Здесь же приведена та информация, которая используется для связывания каналов освещаемой многопроцессорной кластерной системы.

Следующим шагом должно быть создание файлов описания для двух реальных физических интерфейсов *eth*0 и *eth*1, в которых указывается, что они входят в состав связанного канала. Файлы *ifcfg-eth*0 и *ifcfg-eth*1 должны иметь содержимое, представленное в табл. 1.

Таблица 1. Содержимое файлов описания сетевого интерфейса

файл ifcfg-eth0	*файл ifcfg-eth1*
DEVICE=eth0	DEVICE=eth1
USERCTL=no	USERCTL=no
ONBOOT=yes	ONBOOT=yes
MASTER=bond0	MASTER=bond0
SLAVE=yes	SLAVE=yes
BOOTPROTO=none	BOOTPROTO=none

Далее проводится этап инициализации сетевого интерфейса при помощи команды

```
/etc/rc.d/init.d/network start
```

Если дистрибутив системы не позволяет применять *master/slave* нотификацию, то при конфигурации сетевых интерфейсов придется запускать интерфейс связанного канала вручную, используя следующую последовательность команд:

```
/sbin/ifconfig bond0 192.168.1.*
up netmask 255.255.255.0
/sbin/ifenslave bond0 eth0
/sbin/ifenslave bond0 eth1
```

Чтобы каждый раз не выполнять приведенные команды вручную, рекомендуется записать их в какой-нибудь *startup*-скрипт, например, в */etc/rc.d/rc.local*, или заменить ими ту часть скрипта */etc/rc.d/init.d/network*, которая ответственна за инициализацию сетевого интерфейса.

Для ручного запуска сетевого интерфейса мы рекомендуем использовать команду *ifenslave*, которая была разработана в рамках проекта *Beowulf*. Пользователь может ее скомпилировать из исходных кодов, которые представлены непосредственно на сайте проекта *Beowulf* [http://beowulf.org/software/ifenslave.c]. Компиляция этой программы осуществляется следующей командой:

```
gcc -Wall -Wstrict-prototypes -O -
I/usr/src/linux/include ifen-
slave.c -o ifenslave
```

Полученный скомпилированный файл необходимо скопировать в папку */usr/sbin*.

Если по каким-то причинам необходимо, чтобы все сетевые драйверы были инициализированы до загрузки *bonding*-драйвера, следует добавить строку

```
probeall bond0 eth0 eth1 bonding
```

в файл */etc/modules.conf*. Эта инструкция укажет системе, что в случае инициализации интерфейса *bond*0 утилита *modprobe* должна сначала загрузить драйверы для всех сетевых интерфейсов.

Таким образом, настройка технологии *channel bonding* завершается на этом этапе. Если сетевой интерфейс инициализировался без ошибок, это можно проверить, используя команду *ifconfig*. Запустив ее без параметров, пользователь увидит на экране терминала сообщения следующего характера:

```
[root]# /sbin/ifconfig
bond0     Link encap:Ethernet  HWaddr 00:C0:F0:1F:37:B4
          inet addr:192.168.1.*  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:7224794 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3286647 errors:1 dropped:0 overruns:1 carrier:0
          collisions:0 txqueuelen:0

eth0      Link encap:Ethernet  HWaddr 00:C0:F0:1F:37:B4
          inet addr:192.168.1.*  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:3573025 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1643167 errors:1 dropped:0 overruns:1 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:10 Base address:0x1080

eth1      Link encap:Ethernet  HWaddr 00:C0:F0:1F:37:B4
          inet addr:192.168.1.*  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:3651769 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1643480 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:9 Base address:0x1400

lo        Link encap:Local Loopback
                inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Если на экран терминала выводятся сообщения приведенного характера, то можно отметить, что связанный канал успешно сконфигурирован. Как видно, *IP*- и *MAC*-адреса всех сетевых интерфейсов в приведенном варианте получились одинаковыми. Чтобы *switch* мог нормально работать с таким каналом, необходимо настроить режим *Link Aggrigation*. Ознакомиться с тем, каким образом реализуется такая процедура, можно в документации коммутатора. Для разных моделей коммутаторов и разных версий их программного обеспечения такая операция может настраиваться по-разному. В этой связи мы опустим здесь вопросы настройки *Link Aggrigation* на коммутаторах.

Иногда встречаются сообщения (http://studentbank.ru/ view.php?id=9653&p=2), что в некоторых случаях после инициализации виртуального сетевого интерфейса дополнительные каналы не могут сразу принимать входящие пакеты. Такая ситуация может возникнуть по той причине, что новый *MAC*-адрес дополнительных каналов физически не прописывается в *EPROM* сетевой карты. В результате при старте модуля свитч не знает, что этот *MAC*-адрес присоединен к более, чем одному порту. Для того, чтобы сообщить свитчу правильный набор *MAC*-адресов, достаточно непосредственно после инициализации интерфейса выполнить несколько пингов. После того, как *ICMP*-пакеты пройдут через коммутатор по всем виртуальным каналам, внутренняя таблица коммутатора примет правильный вид, и в дальнейшем проблем с приемом пакетов не будет.

В соответствии с приведенной методикой осуществляется инициализация сетевых интерфейсов и остальных узлов кластерной системы.

Исследование основных сетевых характеристик для режима агрегации каналов сетевого интерфейса многопроцессорной системы

С учетом применения технологии *channel bonding*, на первом этапе исследований определим основные сетевые характеристики кластерной системы. Коэффициент пропускной способности сети кластера будем определять следующем образом:

$$k_s = \frac{V_p \cdot N \cdot k \cdot d}{V_b \cdot k_m}. \qquad (1)$$

Здесь приняты следующие обозначения: V_p – протокольная пропускная способность сети кластера, Гбит/с; d – полудуплексный (d = 1) или

дуплексный (d = 2) режим работы вычислительной сети кластерной системы;

N – число узлов кластера; V_b – пропускная способность используемого коммутатора (V_b), Гбит/с; k – количество симметричных вычислительных подсетей, которые работают одновременно за счет реализации технологии *channel bonding*, k_m – количество коммутационных матриц в сети обмена данных.

Коэффициент пропускной способности коммутатора (k_b) уточним аналогично:

$$k_b = \frac{V_b \cdot k_m}{V_p \cdot N \cdot k \cdot d}. \qquad (2)$$

Кроме того, для анализа согласования выбранной коммутационной шины с возможностями коммутатора определим коэффициент полосы пропускания коммутатора (c_k), который с учетом режима агрегации каналов сетевого интерфейса будет определяться соотношением вида:

$$c_k = \frac{V_b \cdot k_m}{N}. \qquad (3)$$

Исходные данные для изучения рассматриваемого режима работы сетевого интерфейса многопроцессорной системы перечислены в табл. 2.

Таблица 2. Исходные данные для расчета сетевых характеристик кластерной системы

V_p	1 Гбит/с
V_b	24 Гбит/с
k	2
k_m	2

На первом этапе исследований выведем аналитическое соотношение для определения равновесного числа узлов кластерной системы [6]. Для этой цели приравняем коэффициенты $k_s = k_b$:

$$\frac{V_p \cdot N \cdot k \cdot d}{V_b \cdot k_m} = \frac{V_b \cdot k_m}{V_p \cdot N \cdot k \cdot d}. \qquad (4)$$

После некоторых преобразований соотношения (4) получаем квадратное уравнение, требуемое значение корня которого будет определяться соотношением вида:

$$N = \frac{V_b \cdot k_m}{V_p \cdot k \cdot d}. \qquad (5)$$

Анализ соотношения (5) показывает, что равновесное число узлов кластерной системы в режиме агрегации каналов сетевого интерфейса, при равных прочих условиях, зависит еще и от количества симметричных вычислительных подсетей (k), которые работают одновременно за счет реализации технологии *channel bonding*, а также и от количества коммутационных матриц в сети обмена данных (k_m).

С учетом заявленных возможностей сетевого интерфейса (табл. 2) на основании соотношения (5) определим равновесное число узлов кластерной системы, которое соответствует N = 12.

Далее для уточнения особенностей функционирования сетевого интерфейса кластерной системы на основании соотношений (1) – (3) была проведена процедура моделирования основных его числовых характеристик.

При этом было установлено, что равновесное число узлов кластерной системы (N = 12), а значение полосы пропускания коммутатора c_k = 4 Гбит/с.

Итак, появились предпосылки для общего анализа полученных результатов. Очевидно, что представленный режим работы, при равных прочих условиях, за счет изменения архитектуры сетевого интерфейса многопроцессорной системы позволяет расширять только полосу пропускания коммутационной шины. *Последнее обстоятельство означает, что сформиро-ванный режим работы сетевого интерфейса кластерной системы будет предоставлять более широкие возможности для реализации процедуры обмена данными между вычислительными узлами, существенно улучшая характеристики эффективности, быстро-действия и надежности функционирования системы.*

Исследование оценок эффективности кластерной системы для режима агрегации каналов сетевого интерфейса

На этом этапе исследований будут выявлены аналитические зависимости для определения числовых характеристик эффективности и ускорения вычислений кластерной системы за счет расширения возможностей сетевого интерфейса. На последующем этапе исследований будут рассмотрены особенности взаимодействия сетевого интерфейса кластерной системы с ее узлами.

Исходные данные для изучения оценок эффективности многопроцессорной системы перечислены в табл. 3.

Таблица 3. Исходные данные для расчета оценок эффективности кластерной системы

V_p	1 Гбит/с
T_{it}	100 с
R	8 Гбит
m	2
d	2
k	2

Здесь приняты следующие обозначения: T_{it} – время счета одной итерации относительно области вычислений, с; R – объем оперативной памяти узла кластера, Гбит; значение m может равняться единице для одностороннего режима граничного обмена данными, или двум для двустороннего.

Для рассматриваемой кластерной многопроцессорной системы в условиях рассматриваемого эксперимента оценим количество узлов кластерной системы, при котором задача будет решаться наиболее эффективно.

В работе [7] выведено уравнение относительно числа узлов N для определения оптимального числа узлов кластерной системы, при котором общее время вычислений, требуемое для решения задачи, будет минимальным. Такое уравнение имеет вид:

$$N^2 - N - \frac{T_{it} \cdot k \cdot d \cdot V_p}{m \cdot \sqrt{R}} = 0 . \qquad (6)$$

Решением уравнения (6) будут два корня, при этом один из них положительный, а другой – отрицательный. Исходя из поставленных физических условий задачи, принимается положительный корень, значение которого равно девяти, т.е. $N = 9$.

Выполним анализ полученного результата. Моделирование основных сетевых характеристик кластерной системы показало, что для оптимального числа узлов кластерной системы ($N = 9$) коэффициент пропускной способности сети кластера $k_s = 0{,}75$, а коэффициент пропускной способности коммутатора k_b = 1,33. При этом

$$k_s < k_b . \qquad (7)$$

Итак, с учетом приведенных значений коэффициентов сети можно уточнить особенности функционирования сетевого интерфейса кластерной системы.

В рамках анализа работы сетевого интерфейса кластерной системы на первом этапе отметим некоторые особенности функционирования коммутатора. При этом заметим, что производительность коммутатора существенно зависит от типов коммутации. Используемый коммутатор поддерживает два основных типа коммутации:

– сквозная коммутация (*cut-through*);

– коммутация с буферизацией (*store-and-forward switching*);

С учетом неравенства (7) можно отметить, что коммутатор будет работать в режиме сквозной коммутации. При сквозной коммутации в буфер входного порта поступают лишь несколько первых байтов пакета, что необходимо для считывания адреса назначения. После установления адреса назначения, параллельно с приемом остальных байтов кадра, происходит коммутация необходимого маршрута, и пакет передается к выходному порту, если он не используется другими устройствами кластера. В противном случае, весь пакет поступает в буфер входного порта. Сквозная коммутация обеспечивает самую высокую скорость коммутации, что дает значительный выигрыш в производительности.

Далее, на основании соотношений, выведенных в работе [7], было проведено моделирование основных характеристик эффективности многопроцессорной системы. Полученные результаты сведены в табл. 4.

Таблица 4. Результаты расчета основных характеристик эффективности при реализации двухканального режима функциони-рования вычислительной сети кластера

Колич. узлов, N	T_n	T_{ex}	T	*USK*	*EF*
1	100,00	0,00	100,00	1,00	1,00
2	50,00	1,41	51,41	1,94	0,97
3	33,33	2,83	36,16	2,77	0,92
4	25,00	4,24	29,24	3,42	0,85
5	20,00	5,66	25,66	3,90	0,78
6	16,67	7,07	23,74	4,21	0,70
7	14,29	8,49	22,77	4,39	0,63
8	12,50	9,90	22,40	4,46	0,56
9	11,11	11,31	22,42	4,46	0,50
10	10,00	12,73	22,73	4,40	0,44
11	9,09	14,14	23,23	4,30	0,39
12	8,33	15,56	23,89	4,19	0,35
13	7,69	16,97	24,66	4,05	0,31
14	7,14	18,38	25,53	3,92	0,28
15	6,67	19,80	26,47	3,78	0,25

Итак, имеем предпосылки для количественной оценки эффективности многопроцессорной системы при реализации двухканального режима функционирования вычислительной сети кластера ($к = 2$). В рамках рассматриваемой задачи оптимальное число узлов кластерной системы, при котором достигается максимальная эффективность распараллеливания, будет соответствовать $N = 9$. При выбранном размере кластера задача будет решаться в 4,6 раза быстрее, чем на одном компьютере. Как показывают расчетные данные, такой режим работы кластера позволил не только повысить эффективность системы, но и существенно сократить время вычислений. Так, время вычислений уменьшилось с 30,81 с при одноканальном режиме функционирования сетевого интерфейса с до 22,4 с.

Исследование загрузки вычислительной сети кластерной системы для режима агрегации каналов сетевого интерфейса

Рассмотрим характеристику коэффициента использования сети кластерной системы для режима агрегации каналов сетевого интерфейса. Такая характеристика необходима для проверки правильности подобранного сетевого оборудования. С этой целью выведем соотношение для коэффициента использования сети через параметры кластерной системы. Значение коэффициента использования сети можно записать в виде следующего аналитического соотношения:

$$\xi = \frac{m \cdot N \cdot (N-1) \cdot \sqrt{R}}{T_i \cdot k \cdot d \cdot V_p + N \cdot m \cdot (N-1) \cdot \sqrt{R}}. \qquad (9)$$

Результаты расчета коэффициента использования сети для дуплексного режима работы многопроцессорной системы показали, что для оптимального числа узлов кластерной системы (N = 9) коэффициент использования сети кластера равен $\xi = 0{,}5$.

Анализ соотношения (9) позволяет сделать вывод, что, как и ожидалось, при увеличении числа узлов кластерной системы значение коэффициента использования сети будет расти. С другой стороны, известно [8], что для сетевой технологии *Ethernet* при ξ = 50 % оперативная память коммутатора будет использоваться приблизительно на 70 %. Запас этой памяти (до 30 %) резервируется для устранения коллизий, которые могут возникать в результате загруженности вычислительной сети кластера. При этом сеть многопроцессорной системы будет работать в режиме сквозной коммутации.

Таким образом, при загрузке сети до 50 % технология *Ethernet* на разделяемом сегменте хорошо справляется с передачей трафика, генерируемого узлами много-процессорной системы. Однако при повышении интенсивности генерируемого трафика сеть начинает обрабатывать данные неэффективно, повторно передавая кадры, которые вызвали коллизию. При возрастании интенсивности генерируемого трафика возникает ситуация, при которой практически любой кадр, который пытается передать некоторый узел многопроцессорной системы, сталкивается с другими кадрами, вызывая коллизию. Сеть перестает передавать полезную информацию и работает "на себя", обрабатывая коллизии.

Полученные результаты позволяют сделать вывод, что с одной стороны, как и было, установлено ранее, при выбранном режиме функционирования кластера можно использовать не больше девяти лезвий, а, с другой стороны, можно отметить, что оборудование сетевого интерфейса кластерной системы подобрано удачно.

Заключение

В статье показаны пути повышения эффективности многопроцессорной кластерной системы за счет реорганизации архитектуры ее сетевого интерфейса. Предложенный подход позволил не только повысить эффективность распараллеливания, но и существенно уменьшить время вычислений. Таких результатов удалось достичь за счет уменьшения времени граничного обмена данных между вычислительными узлами кластерной системы.

Показано, что главное преимущество режима агрегации каналов состоит в том, что существенно повышается скорость обмена данными. При этом основная особенность такого режима работы кластерной системы состоит в том, что повышается надежность ее функционирования. Так, в случае отказа адаптера трафик посылается следующему работающему адаптеру без прерывания сервиса. Если же адаптер вновь начинает работать, то через него опять пересылаются данные.

Список использованной литературы

1. Модуль високоефективної багатопроцесорної системи підвищеної готовності: пат. 57663 Україна, МПК G06F 15/16 (2011.01) / Іващенко В.П., Башков Є.О., Швачич Г.Г., Ткач М.О.; власники: Національна металургійна академія України, Донецький національний технічний університет. – № u 2010 09341; заявл. 26.07.2010; опубл. 10.03.2011, Бюл. № 5.

2. Башков Є.О. Високопродуктивна багатопроцесорна система на базі персонального обчислювального кластера / Є.О. Башков, В.П. Іващенко, Г.Г. Швачич // Наукові праці Донецького національного технічного університету. Серія «Проблеми моделювання та автоматизації проектування». – 2011. – Вип. 9 (179). – С. 312 – 324.

3. Швачич Г.Г. Математическое моделирование скоростных режимов термической обработки длинномерных изделий / Г.Г. Швачич, В.П. Колпак, М.А. Соболенко // Теория и практика металлургии. Общегосударственный научно-технический журнал. – 2007. – № 4 – 5 (59 – 60). – С. 61 – 67.

4. Швачич Г.Г. Про проблему математичного моделювання термічної обробки довгомірного сталевого виробу / Г.Г. Швачич, М.О. Ткач // VII Intrenational Conference "Strategy of Quality in Industry and Education", June, 3 – 10. 2011. – Varna; Bulgaria. – Proceedings. – V. 2. – P. 561 – 567.

5. Установка для термічної обробки довго вимірного сталевого виробу: пат. 61944 Україна, МПК C21D 1/26 (2006.01) G06F 15/16 (2006.01) / Іващенко В.П., Башков Є.О., Швачич Г.Г., Ткач М.О.; власники: Національна металургійна академія України, Донецький національний технічний університет. – № u 2010 14225; заявл. 29.11.2010; опубл. 10.08.2011, Бюл. № 15.

6. Кластерные решения [Электронный ресурс]. – Режим доступа: http://www.hardline.ru/2/22/1559.

7. Исследование влияния сетевого интерфейса на эффективность модульной многопроцессорной кластерной системы / Е.А. Башков, В.П. Иващенко, Г.Г. Швачич и др. // Наукові праці Донецького національного технічного університету. Серія «Інформатика, кібернетика та обчислювальна техніка». – 2011. – Вип. 14 (188). – С. 89 – 99.

8. [Электронный ресурс]. – Режим доступа: http://kafvt.narod.ru/Osia/Glava4.htm.

Надійшла до редколегії 21.03.2012

Є.О. БАШКОВ [1], В.П. ІВАЩЕНКО [2], Г.Г. ШВАЧИЧ [2]

[1]Донецький національний технічний університет, м. Донецьк, Україна,

[2]Національна металургійна академія України, м. Дніпропетровськ, Україна

E.A. BASHKOV [1], V.P. IVASHCHENKO [2], G.G. SHVACHYCH [2]

[1]Donetsk national technical university, Donetsk, Ukraine

[2]National metallurgical academy of Ukraine, Dnepropetrovsk, Ukraine

РЕАЛІЗАЦІЯ РЕЖИМУ АГРЕГАЦІЇ КАНАЛІВ МЕРЕЖЕВОГО ІНТЕРФЕЙСУ В МОДУЛЬНИХ БАГАТОПРОЦЕСОРНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

Статтю присвячено проблемі підвищення ефективності багатопроцесорних кластерних систем за рахунок реорганізації архітектури її мережевого інтерфейсу. Запропонований підхід дозволив істотно зменшити час граничного обміну даних між обчислювальними вузлами кластерної системи, що дає можливість не лише підвищити ефективність розпаралелювання, але і суттєво зменшити час обчислень.

Ключові слова: багатопроцесорна обчислювальна система, мережевий інтерфейс, ефективність розпаралелювання, прискорення обчислень, комутатор.

REALIZATION OF MODE OF AGGREGATING OF DUCTINGS OF NETWORK INTERFACE IS IN MODULE MULTIPROCESSOR COMPUTER SYSTEMS

The article is devoted to the problem for increasing of the multiprocessor cluster system's efficiency at the expense of reorganization touching upon the architecture of its network interface. The suggested approach allowed to decrease substantially the time of boundary exchange of data between calculating units of a cluster system, that gives opportunity not to increase the efficiency of disparalleling but also to decrease substantially the time of calculations.

Keywords: multiprocessor computer system, network interface, efficiency of disparalleling, acceleration of calculations, switchboar.