DOI: 10.36910/6775-2524-0560-2020-39-28
УДК 004.451
Мельник Василь Михайлович, к.ф.-м.н., доцент,
http://orcid.org/0000-0001-8282-6639
Мельник Катерина Вікторівна, к.т.н., доцент,
http://orcid.org/0000-0002-9991-582X
Кузьмич Олена Іванівна, к.т.н., доцент,
http://orcid.org/0000-0002-8717-4497
Багнюк Наталія Володимирівна, к.т.н., доцент,
https://orcid.org/0000-0002-7120-5455
Кравець Олег Русланович, магістрант
Луцький національний технічний університет

ДОСЛІДЖЕННЯ ПОКРАЩЕННЯ ВНУТРІШНІХ ТА ЗОВНІШНІХ ПАРАМЕТРІВ ШВИДКОДІЇ ЗВ'ЯЗКУ НА КЛАСТЕРІ КОМУНІКУЮЧИХ ВІРТУАЛЬНИХ МАШИН

В.М. Мельник, К.В. Мельник, О.І. Кузьмич, Н.В. Багнюк, О.Р. Кравець. Дослідження покращення внутрішніх та зовнішніх параметрів швидкодії зв'язку на кластері комунікуючих віртуальних машин. На кластері віртуальних машин для високопродуктивної обробки даних, який є бінарно сумісним для додатків зі стандартним сокетним інтерфейсом, виявлено покращення внутрішніх та зовнішніх параметрів мережевої швидкодії за допомогою введеного в систему механізму спрощеної комунікації, реалізованого на базі Xen 3.2 та ядра Linux, який демонструє взаємодію віртуальних машин, подібно до організації зв'язку на основі UNIX DOMAIN сокетів. Встановлено, що пропускна здатність між комунікуючими машинами з використанням спрощеного зв'язку зростає приблизно на 2,11 %, ніж для традиційного TCP/IP протоколу, а швидкодія передачі повідомлень – на 7,8–7,9 %.
Ключові слова: кластер віртуальних машин, механізм спрощеної комунікації, високопродуктивна обробка, параметри швидкодії.

В.М. Мельник, К.В. Мельник, О.И. Кузьмич, Н.В. Багнюк, А.Р. Кравец. Исследование улучшение внутренних и внешних параметров быстродействия связи на кластере коммуницирующих виртуальных машин. На кластере виртуальных машин для высокопроизводительной обработки данных, который является бинарно совместимым для приложений со стандартным сокетных интерфейсом, выявлено улучшение внутренних и внешних параметров сетевой производительности с помощью введенного в систему механизма упрощенной коммуникации, реализованного на базе Xen 3.2 и ядра Linux, который демонстрирует взаимодействие виртуальных машин, подобно организации связи с помощью UNIX DOMAIN сокетов. Встановлено, что пропускная способность между коммуницирующими машинами с использованием упрощенной связи возрастает примерно на 2,11%, чем для традиционного TCP/IP протокола, а быстродействие передачи сообщений – на 7,8-7,9%.
Ключевые слова: кластер виртуальных машин, механизм упрощенной коммуникации, высокопроизводительная обработка, параметры быстродействия.

V. Melnyk, K. Melnyk, O. Kuzmych, N. Bahniuk, O. Kravets. Internal and external communication performance improving research on a cluster of communicated virtual machines. improvements to the internal and external parameters of network performance are revealed on the virtual machine cluster for high-performance data computing that is binary compatible for applications using the standard socket interface with the simplified communication mechanism introduction implemented on the basis of Xen 3.2 and Linux kernel, which demonstrates the interoperability between machines with similar to UNIX DOMAIN sockets communication. It is revealed that bandwidth between communication machines with simplified communication is increases by about 2.11%, than for traditional TCP/IP protocol, and the speed of message transmission are increased on 7.8-7.9%.
Keywords: virtual machines cluster, simplified communication mechanism, high-performance processing, performance parameters.

1. Постановка проблеми

Для вирішення завдань високої складності в наш час використання високопродуктивних обчислень набуло високого розмаху. Подібні обчислення здійснюються не тільки в суто науковій діяльності природничих наук, як це спостерігалось ще недавно. Вони актуальні і набули сьогодні великого розмаху в різних сферах людської діяльності [1-4]. Однак, з аналізу пророблених нами літературних даних [5–8], їх продуктивність ще може зазнавати вагомих втрат від різноманітних недоліків технічного походження. Наприклад, спад продуктивності обчислень може проявлятися в залежності від низької пропускної здатності мережі та її продуктивності. Така залежність може спостерігатися як збоку зовнішньої швидкодії передачі даних, так і всередині кластера високопродуктивних обчислень. Такий спад може бути обумовлений і ростом часу затримки передачі повідомлень.

Для того, щоб вирішити вище описані недоліки, багатьма авторами наукових статей було здійснено спробу розробки та реалізації багатьох технічних та програмних механізмів. Так, з метою покращення міждоменної комунікації в роботі [5] було запропоновано використання ком-сокетів, які повинні б значною мірою обумовити зростання таких мережевих параметрів, як пропускної здатності

та зниження часу затримки отримання повідомлення. Авторами даної роботи був змодельований і практично втілений спеціальний підхід для спрощення механізму комунікації і покращення продуктивності обміну. З метою вирішення задачі швидкодії обміну даними між хмарами у наукових медичних цілях в роботі [6] було залучено практичний підхід з використанням java-сокетів та формування єдиної копії протоколу під час повторного використання ниток виконання. В даній роботі використовувалися бібліотеки спеціально розроблені для підвищення швидкості обміну повідомленнями по мережі, які, відповідно, сприяли покращенню і інших мережевих характеристик впливу на процес обміну даними та зростання його продуктивності.

В інших роботах [16] увага приділялася питанням вивчення впливу інших типів сокетів на продуктивність обробки даних на базі різноманітних комп'ютерних систем та механізми їх захисту. В деяких із них для вивчення параметрів продуктивності під час взаємодії java-додатків з мережею впроваджувались спеціальні механізми-аналізатори. У решти зазначених робіт для високопродуктивних обчислень та дослідження з'єднання зосереджувалася увага на вивченні мережевих параметрів затрат, використанні займаної пам'яті та надійності передавання даних по мережі. Проте домінуючим із найбільш актуальних в роботах вирішувалося питання підвищення параметрів продуктивності чи пропускної здатності різних типів мережі на різних ділянках її функціонування.

Досить вагомо в наші дні зростає інтерес і до високопродуктивних хмарних обчислень, а також побудови швидкісних обчислювальних систем які також активно впроваджуються в різні види громіздких обчислень для вирішення важких завдань у різних сферах людської діяльності [7,8]. В наведених наукових працях також вирішується питання підвищення продуктивності мережі з метою обміну необхідною для обробки інформацією між додатками, які з'єднуються з хмарами, що містять різноманітні медичні дані, встановлені в результаті діагностики та аналізів у різних клініках чи діагностичних центрах, а також встановлених заключень та історій захворювань під час перебування на лікуванні в стаціонарних лікарнях. Використовуючи java-сокети, було здійснено спробу вирішення задачі підвищення швидкої обробки вищезгаданих даних та роботоздатності додатків з хмарами в цілому. Для вирішення подібних проблем обробки різнобічної інформації все частіше спеціалістам з різних галузей рекомендуються програмні системи обробки у вигляді додатків, які призначені для хмарних обчислень, для яких є необхідним високошвидкісний інструмент управління доступом до даних та їх передачею по мережі. В зазначених працях заодно вирішується також питання ефективного використання задіяних ресурсів для проведення високопродуктивної обробки даних.

2. Аналіз досліджень

На основі проведених досліджень та отриманих результатів в роботах [9–11] було конкретно відзначено, що гостьовий Xen-домен в Linux працює зі значно нижчою продуктивністю в мережі комунікації з хмарою, ніж власний домен Linux. Подібна проблема вирішується також в роботі [12], тільки в умовах, коли додаток, встановлений на віртуальній машині, комунікує з іншою віртуальною машиною, яка встановлена на іншій фізичній машині. У результаті пророоблених дослідженнях було виявлено зниження продуктивності системи з досить помітним коефіцієнтом 2–3 по відношенню до вхідного її навантаження. Ще більше зниження продуктивності з коефіцієнтом 5 було виявлене і для вихідного навантаження на мережу з'єднання. В даних роботах констатується також, що TCP/IP сімейство протоколів, розроблене для комплексної комунікації в середовищі мережі Інтернет досить не підходить для продуктивної роботи в середовищі кластера віртуальних машин.

В роботі [16] наведено модель взаємодії двох типів зразків з'єднання між віртуальними машинами в кластері. Згідно даних запропонованої в роботі моделі для реалізації внутрідоменного зв'язку бралися до уваги віртуальні машинами 1 і 2, які були встановлені на одній і тій же фізичній машині. Позадоменний зв'язок згідно моделі було втілено між віртуальними машинами 3 і 4, які були встановлені на інших окремих фізичних машинах. Для вирішення зазначеної проблеми підвищення продуктивності в роботі запропоновано також розроблений механізм спрощеної комунікації для мережевої комунікації між віртуальними машинами, які входили в склад віртуального кластера. Заодно, в експерименті ставилася задача і перевірки працюючих на цих машинах додатків на повну бінарну сумісність на основі використання стандартних сокетів та їх інтерфейсу і роботі.

Для втілення поставленої задачі продуктивності мережі та зростання швидкодії передачі повідомлень був впроваджений протокол спрощеного зв'язку, який володів низьким часом затримки передавання повідомлень. Перевага його становила і те, що розробникам зовсім не потрібно було переписувати працюючі додатки, які брали участь в комунікації з хмарою. Скорочена модель взаємодії віртуальних машин всередині кластера та позакластерна взаємодія машин зображені на рисунку 1. Запропонований в роботі протокол використовувався виключно для удосконалення

внутрішнього доменного зв'язку, а не позадоменного, що також був локалізований для з'єднання з віртуальним кластером. Іншими словами, для будь-якої віртуальної машини, що входить у склад віртуального кластера і підтримує робочий зв'язок з фізичною або віртуальною машиною поза його межами, необхідними є традиційні TCP/IP протоколи.

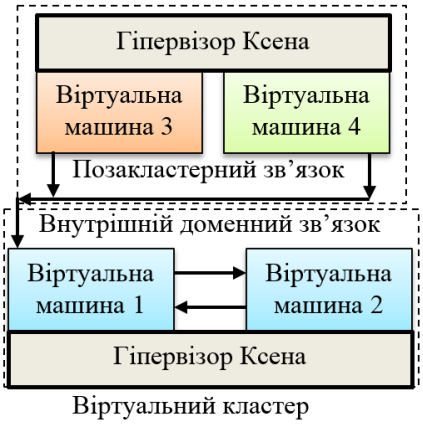

Рис. 1 – Модель комунікації віртуальних машин у кластері та позакластерний зв'язок

Для поліпшення продуктивності міждоменної мережі також потрібно було інтегрувати петлю Ксена згідно моделі роботи [13] в машині зі спрощеною комунікацією зв'язку. Для отримання кращої продуктивності в з'єднаннях міждоменної мережі, петля Ксена повинна здійснювати перехоплення вихідних пакетів повідомлень ще перед мережевим рівнем згідно моделі TCP/IP. Далі її робота полягає у виявленні тих повідомлень, які призначені для обчислювальних робіт на машинах віртуального кластера. Слід додати ще, що ці віртуальні машини об'єднані високошвидкісним каналом розділювальної пам'яті, який здійснює обхід традиційного інтерфейсу віртуальізованої мережі.

Під час аналізу літературних даних були окремі роботи, дослідження в яких базувалися на зменшенні накладних затрат, що відносяться до мережевої віртуалізації вводу/виводу та впливали на роботу і продуктивність обробки даних хмарних додатків. Зменшення мережевих затрат обумовлюється застосуванням високошвидкісного каналу зв'язку між окремими робочими доменами. В роботі [14] підвищення мережевої продуктивності обумовлене обходом операцій з залученням TCP/IP-стеків, під час яких мають прояв затрати на гортання сторінок. Обходячи їх забезпечується пряме прискорення каналу з'єднання між віртуальними доменами, розміщеними на одній і тій же фізичній машині. Використання високопродуктивного міждоменного механізму зв'язку XWay здійснює перехоплення зв'язаних викликів ще між рівнями INET та TCP. Механізм XWay застосовує спеціальний модуль комутації для визначення місця знаходження домена призначення. Якщо домен знаходиться на одній і тій же фізичній машині, то комутуючий перемикач намагається відкрити високошвидкісний XWay-канал зв'язку, поєднавши його з XWay-сокетом. Таким чином, з рівня комутатора запит INET пересилається на TCP-рівень, причому це зовсім не впливає на зовнішній зв'язок в мережі, тому що для неї XWay-канал не існує. Треба сказати також, що XWay підтримує в роботі повну бінарну сумісність додатків, що запрограмовані для сокетів зі традиційним інтерфейсом. Та все ж цей механізм призначений виключно для організації зв'язку всередині домена. Поряд з тим, багато мережевих додатків для інтенсивної обробки даних у кластерах віртуальних машин використовують не позадоменний зв'язок, а беруть за основу модель гібридної внутрідоменної комунікації [15]. В такому разі за законом Амдаля здійснення оптимізації з'єднання тільки виключно між доменами не приведе до значного зростання продуктивності такої системи [11].

Відповідно до закону GPL [13] механізм XenLoop представляється у вигляді звичайного програмного проекту з доступним вихідним кодом. Так само, як і XWay, механізм XenLoop напрямлений на підтримку підвищення продуктивності мережі всередині домена в середовищі об'єднаних віртуальних машин. На рівні користувача він сильно акцентує на питанні доступності до віртуальних машин і домагається порівняно вищої продуктивності між гостьовими машинами-співрезидентами кластера за рахунок покращення з'єднання. Його дія в основному полягає в аналізі та перевірці мережевих пакетів, які надходять на мережевий рівень, і відслідковує їх, які із них призначені для отримання і якими віртуальними машинами, враховуючи розміщення їх на одній фізичній машині. Така перевірка виконується через використання високошвидкісного каналу

комунікації, організованого за допомогою спеціально створеної розділювальної пам'яті між взаємодіючими віртуальними машинами, зовсім обминаючи традиційний мережевий інтерфейс віртуалізації. Таким чином, віртуальна машина-гість за допомогою введеного механізму XenLoop може вільно мігрувати між об'єднаними машинами, зовсім не торкаючись діючої мережевої комунікації та зовсім доступно виконувати і підтримувати з'єднання між діючим каналом XenLoop та мережевим розгалуженням комунікації кластера.

Механізм XenLoop базується на побудові програмних модулів для Linux-ядра в домені Xen з індексом 0 та інших взаємодіючих гостьових доменах. Будь-яка віртуальна машина у ньому в робочому режимі має можливість динамічно завантажувати модуль XenLoop. З проведеного аналізу та оцінки застосування послідовності немодифікуючих тестів підтверджено [13], що дія XenLoop намагається знижувати значення часу затримки обходу повідомлення при проходження ним повного кругового маршруту між взаємодіючими віртуальними машинами приблизно в 5 разів і, відповідно, підвищувати пропускну здатність на ділянці задіяної ділянки мережі приблизно в 6 разів. Проте, як і у випадку використання XWay, за допомогою наведеного XenLoop механізму можна оптимізувати продуктивність комунікації тільки всередині домена, що розцінюється як недолік. Оскільки на мережевому рівні XenLoop здійснює перехоплення вхідних мережевих пакетів, то в роботу обов'язково задіються додаткові ресурси CPU, які і обумовлять зайві затрати на TCP/IP-обробку. Архітектура використання механізму XenLoop в домені користувача описана в роботі [16].

Існує інший спосіб, застосований в роботі [17], який також спрямований до оптимізації мережі з підвищенням її продуктивності. Він полягає в залученні моніторингу до відтворення треку проходження пакетів пересилання між взаємодіючими віртуальними машинами, чи хоча би між привілейованими віртуальними машинами. Його використання дає можливість вагомо покращувати продуктивність зв'язку та безпечно чи відокремлено відтворювати операції вводу-виводу між віртуальними машинами. Архітектуру моніторингу проходження між робочими віртуальними машинами зображено на рисунку 2. Однак, сам моніторинг проходження між ними можна застосовувати тільки в такому віртуальному середовищі, в якому мережеве обладнання надає підтримку проходженню діючої операційної системи. То ж лише прикладні програми, які написані з урахуванням застосування інтерфейсу проходження операційної системи, мають можливість забезпечення продуктивністю загаданого моніторингу обходу між діючими віртуальними машинами.

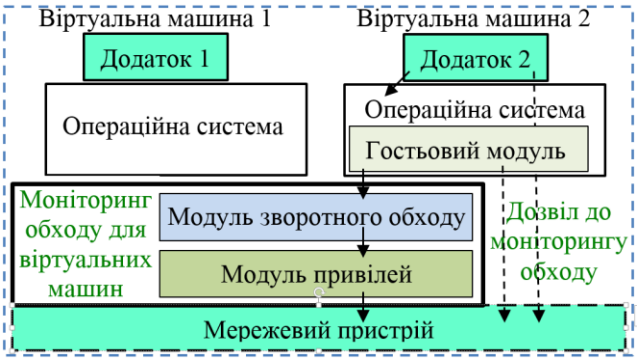

Рис. 2 – Архітектурна схема моніторингу обходу між робочими віртуальними машинами

Протокол віддаленого доступу до пам'яті розроблений у вигляді спеціального надійного датаграмного протоколу віконного типу для реалізації спрощеного зв'язку. Він цілеспрямовано здійснює віддалений доступ до сегмента розділювальної пам'яті, яка належить одній підмережі. Він може бути поданий у вигляді програмного модуля [18, 19], який здійснює координацію використання ресурсів пам'яті в широких кластерних системах з метою підтримки великого її обсягу та втілення інтенсивних навантажень для операцій вводу-виводу. Він слугує абсолютно прозорою розподіленою системою для віртуальних машин в середовищі Xen. В процесі ефективної передачі, наприклад, однієї сторінки пам'яті протокол віддаленого доступу видаляє традиційні TCP-функції, до яких відноситься функціональна абстракція, яка відповідає за потоково-байтовий порядок доставки повідомлень, здійснення контролю перевантажень і функціональності IP-маршрутизації в межах локальної мережі. Таким чином протокол віддаленого доступу до пам'яті виконує обхід традиційного стеку TCP/IP протоколів і напряму з'єднується з драйвером мережевого пристрою. Він умовно налаштовується на передачу сторінкової пам'яті таким способом, щоб інші програми, які здійснюють власну взаємодію через сокети, не отримували покращення продуктивності під час його роботи.

Дослідження було виконано на базі системи Xen 3.2 з ядром Linux 2.6.18. Надалі здійснимо короткий огляд системи з Xen та архітектури виконання операцій вводу-виводу для Xen-мережі, зображеної на рисунку 3, продемонструвавши її особливості побудови. Xen в нашому випадку – це не що інше, як монітор віртуальної машини x86, що дозволяє операційним системам багаторазового спрямування розподіляти поміж собою діюче апаратне обладнання. Використовуючи Xen є можливість запуску декількох операційних систем для одночасної роботи на одній фізичній машині, де кожна операційна система виступає гостьовим доменом, між якими повинен бути один домен 0 з привілейованим хостингом для здійснення управління діючого програмного забезпечення на рівні додатків користувача.

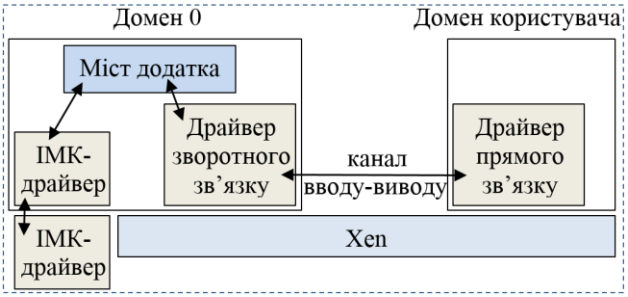

Рис. 3 – Схема архітектури для операцій вводу-виводу в мережі Xen

Дана схема представлення Xen-системи подає два віртуалізаційні підходи. Перший підхід реалізований на основі представленої абстракції діючого апаратного обладнання представляє *пара-віртуалізацію*, яка подібна, однак не ідентична до схем роботи апаратних комплексів, сконструйованих для функціонування операційних систем для гостьових користувачів. А тому деякі з діючих операційних систем, зокрема сімейства Windows, не спроможні працювати в даному режимі роботи. Повна абстракція підлеглого апаратного забезпечення реалізує *повну Xen-віртуалізацію*, проте вона потребує робочої підтримки з процесорної сторони.

Діючі в наш час мережеві комунікації, спроектовані у відповідності до Xen здійснюють свою роботу через використання мережевого інтерфейсу віртуалізації. На рисунку 3 схематично наведений домен користувача, який здійснює з'єднання з головним доменом через власний фронт-драйвер, або *мережевий вхідний драйвер*. В домені 0 для організації комунікації повинен функціонувати драйвер зворотного зв'язку, або *мережевий вихідний драйвер*. в кожному гостьовому домені мережевий інтерфейс входу повинен бути підключений до відповідного інтерфейсу в домені 0 через власний зворотній або вихідний мережевий кінець. Діючі драйвери кінця *виходу* повинні фізично з'єднуватися через програмний міст з мережевою платою. Мережеві додатки поєднані з роботою гостьових операційних систем надсилають свої повідомлення на власний фронт-драйвер, який, в свою чергу, продовжує надсилання повідомлень до зворотного чи вихідного драйвера, відтворюючи операції їх копіювання чи сторінкового гортання.

Проаналізувавши наведені в літературі наукові результати можна зробити відповідний висновок, що питання зменшення додаткових мережевих затрат які мають місце у функціонуванні кластера віртуальних машин та дослідження параметрів швидкодії передачі даних з метою покращення внутрішніх та зовнішніх її параметрів частково вирішувалося тільки деякими науковими роботами. Проте загальне питання дослідження параметрів швидкодії мережі комунікації на різних рівнях саме з метою покращення продуктивності роботи мережевих додатків для високопродуктивних обчислень на віртуальних машинах в хмарі в науковій літературі розглядалося значно менше і на сьогодні становить актуальність подібних наукових досліджень. Також в наш час велика увага приділяється і науковим завданням введення нових ефективних механізмів зв'язку та дослідження їх параметрів швидкодії з метою покращення продуктивності роботи мережевих додатків, враховуючи не тільки внутрікластерну взаємодію, а розглядаючи її в інтеграції з позакластерною та її мережевими ділянками. Додатковим питанням також випливає і бінарна сумісність роботи мережевих додатків, починаючи вже від етапу їх розробки.

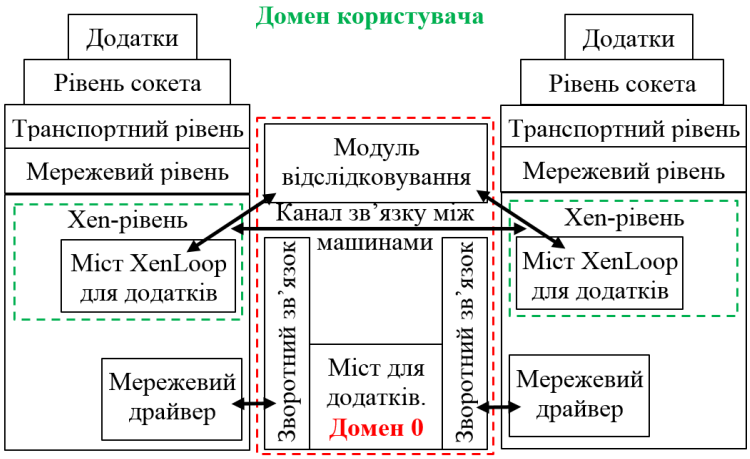

Рис. 4 – Архітектурна схема механізму спрощеної комунікації з Xen

3. Мета і завдання дослідження

Метою даної роботи є дослідження покращення внутрішніх та зовнішніх параметрів швидкодії зв'язку на кластері комунікуючих віртуальних машин на основі впровадженого механізму спрощеного зв'язку в даний хмарний кластер, призначений для виконання високопродуктивних обчислень. Виходячи з даної мети механізм повинен підтримувати бінарну сумісність додатків в процесі їх роботи, не зважаючи на те, що вони створені на основі використання стандартного інтерфейсу для сокетів віртуальних машин.

В дослідженні було намічено наступні завдання для досягнення поставленої мети:

– використавши діючий механізм спрощеного зв'язку, довести кількісне покращення продуктивності мережі у кластері віртуальних машин для виконання хмарних високопродуктивних обчислень. В системі, взятій для проведення дослідження повинен бути використаний розроблений протокол спрощеної комунікації, який би підтримував згадану в меті дослідження бінарну сумісність мережевих додатків у ході їх роботи.

– для покращення продуктивності внутрідоменної комунікації між віртуальними машинами в спроектованій системі впровадити XenLoop-петлю з метою здійснення перехоплення надісланих пакетів повідомлень ще на рівні, нижче мережевого. В процесі дослідження необхідного нам параметра пропускної здатності між двома віртуальними машинами з урахуванням механізму спрощеної комунікації та спеціально розробленого протоколу нам необхідно застосувати еталонний тест з характерним набором NAS-міток і переконатися про проходження наміченого пакета через створений канал FIFO, (див. рис. 4) обумовлений введенням XenLoop-петлі [25].

– в ході проведення дослідження з залученням протоколу спрощеного зв'язку та протоколу віддаленого доступу до пам'яті експериментально довести зростання пропускної здатності в комунікації між доменами, що представлені відповідними віртуальними машинами, яка залежить від міждоменної мережевої пропускної здатності та часу затримки.

4. Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Весь механізм спрощеної комунікації об'єднує в собі два модулі. Перший модуль втілює протокол спрощеного зв'язку, що здійснює управління позадоменним мережевим з'єднанням до кластера віртуальних машин. Другий модуль реалізує рівень *XenLoop*, через який протікає високошвидкісний комунікаційний канал для можливості покращення з'єднання всередині домена між віртуальними машинами у віртуальному кластері [17].

Серед всіх модулів основним є модуль, який відтворює протокол спрощеної комунікації та його роботу. Він містить в собі три об'єднані модулі, перший з яких представляє ділянку ядра. Для перехоплення повідомлень, які надходять зі сторони додатків вищого рівня зі збереженням стабільними програмованих інтерфейсів [20, 21], необхідно в гостьових операційних системах провести незначні зміни, тобто в традиційному ядрі Linux. Інший підлеглий модуль реалізує розроблений протокол спрощеного зв'язку і втілений в змодельовану систему. З літератури відомо, що сімейство TCP/IP протоколів інтегрує в собі адресацію звернень, мультиплексію повідомлень, управління з'єднаннями між кінцевими машинами, контроль шляху проходження, регламентування передачі по лінії зв'язку з метою уникнення перевантажень, фрагментації пакетів з можливістю їх перебудови і інші дії. Він досить добре повинен поєднувати свою роботу в багатокомпонентному мережевому Інтернет-середовищі. Проте, робота протоколу TCP/IP в середовищі віртуального

кластера для високопродуктивних обчислень та роботи з хмарою, в якій інфраструктура мережі володіє набагато вищою продуктивністю порівняно з мережею Інтернет, стає в значній мірі затрудненою для машинних процесорів. На основі цих обставин появилася ідея, що привела до створення протоколу спрощеної комунікації, призначення і робота якого описані в [16]. На діаграмі, зображеній на рисунку 5 зображена логічна послідовність операцій під час встановлення з'єднання з використанням протоколу спрощеного зв'язку.

Рис. 5 – Діаграма логічної послідовності встановлення з'єднання з використанням протоколу спрощеного зв'язку

Наведемо покроковий опис встановлення зв'язку для працюючого додатка в наведеній вище системі з задіяним у роботу протоколом спрощеного зв'язку. Як видно з рисунка 5, діючий додаток, що знаходиться на віртуальній машині, першим кроком викликає сокетну функцію з метою встановлення TCP-з'єднання для обміну повідомленнями. Описана вище надбудова, яка включена в ядро Linux, перехоплює цей виклик і для функції визначення виставляє конкретне завдання виявлення місця знаходження віртуальної машини протилежної сторони зв'язку. В цьому завданні головним є визначити, чи вона знаходиться на тій же самій чи іншій фізичній машині. Функція визначення обов'язково повинна повернути певний результат щодо місця знаходження віртуальної машини протилежної сторони зв'язку. Якщо виявлено, що місцем знаходження віртуальних машин є різні фізичні машини, то за допомогою протоколу спрощеного зв'язку виконується спроба виклику функції встановлення з'єднання. В цей же час здійснюється виділення потрібних структур даних та ресурсів, необхідних для встановлення зв'язку. Виконавши описані процедури, функція встановлення зв'язку намагається повернути очікуваний результат, використовуючи цей же протокол спрощеного зв'язку. У випадку, коли дві гостьові співрезидентні віртуальні машини намагаються встановити активне мережеве з'єднання між собою, створюється динамічний двосторонній канал зв'язку типу FIFO для передачі пакетів між ними, використовуючи відомий нам традиційний протокол "двостороннього рукостискання". Детально описана діаграма, яка висвітлює послідовність зв'язку всередині домена, зображена на рисунку 6. IP-протокол надсилає датаграму повідомлення, яку сепарує призначена функція перехоплення. Використовуючи інформацію модуля відслідковування повідомлень, дана функція визначає приналежність датаграми з використанням для передачі тільки внутрідоменної комунікації. Якщо модулем XenLoop визначається, що датаграма буде використовувати комунікацію всередині домена, то її подальша передача буде виконуватися через високошвидкісний канал між віртуальними машинами, працюючими всередині домена. Звідси також і здійсниться повернення результату про з'єднання.

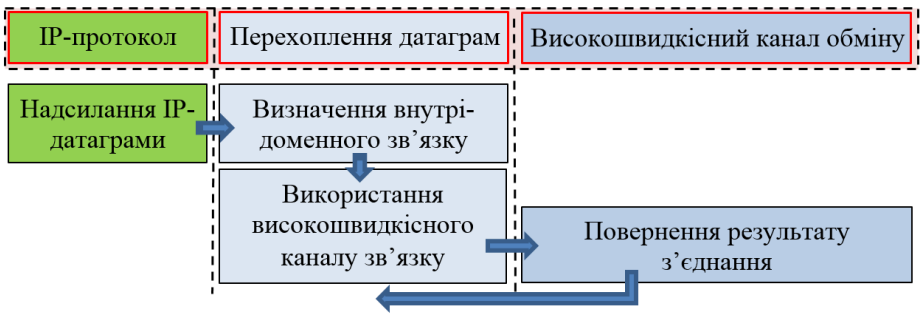

Рис. 6 – Вигляд послідовності комунікації всередині домена

Зупинимося на особливостях роботи модуля, який відтворює протокол спрощеного зв'язку. Всі інші робочі модулі, задіяні в реалізації системи згідно механізмі спрощеної комунікації досить добре представлені в роботі [20]. Модуль, який відтворює протокол спрощеного зв'язку, як і наведений в цій роботі модуль XWay, містить в собі три розглянуті нами програмні модулі, такі як додаток до Linux-ядра, модуль завантаження для ядра та рівня доменного використання. Як уже згадувалося, додаток до ядра працює в сторону виконання бінарної сумісності з іншими сокетними додатками та узгоджується в роботі з модулем виконання завантажень для ядра під час передачі пакетів даних з використанням протоколу спрощеного зв'язку. Програмний модуль завантаження для ядра також включає в себе протокол спрощеного зв'язку, з яким допомагає встановити зв'язок і сам домен на рівні користувача. В додатку до ядра проходить модифікація двох відомих структур ядра Linux, таких як *struct inet_stream_ops* та *struct tcpyrot*, які в традиційному вигляді містять адресні вказівники на функції, які реалізують базові операції контенту виконання для мережі TCP, до яких відносяться з'єднання, відправка, отримання, виключення і інші. Дані вказівники було заміщено вищезгаданими функціями. Кожна із згаданих функцій володіє власним твердженням, яке чітко визначає обробку поточної операції протоколом спрощеного зв'язку чи спеціальною функцією ядра Linux. У випадку використання спрощеного зв'язку вона буде оброблена протоколом спрощеного зв'язку. В іншому випадку операція повинна оброблятися звичайним традиційним TCP/IP-протоколом. Детальний опис програмних структур та їх застосування для відтворення комунікації описані в [16].

Будь-яка датаграма протоколу спрощеного зв'язку містить набір полів в спеціально організованому порядку. Машина-одержувач повинна знати, як потрібно вести зчитування і розшифрування отриманого потоку даних. Заголовок датаграми протоколу спрощеного зв'язку, формат якої показаний на рисунку 7, містить вісім основних полів. Протокол спрощеного зв'язку володіє двома типами датаграм: датаграма розміщення даних та датаграма контрольної інформації для надісланого пакета. Останнє поле номера каналу зберігає номер буферного каналу для активізації з'єднання з використанням протоколу спрощеного зв'язку. Для забезпечення надійної передачі датаграм система спрощеного зв'язку використовує спеціально розроблений віконний алгоритмом передачі повідомлень, описаний в [22].

Контрольна сума	**Номер послідовності**	**Початковий індекс**	**Довжина корисної частини датаграми**
Порядковий номер підтвердження	Індексний номер підтвердження	Тип	Номер каналу
Корисна частина датаграми			

Рис. 7 – Формат датаграми для протоколу спрощеного зв'язку

Оцінимо вплив введеного в запропоновану систему *механізму спрощеного зв'язку* [23] на основі отриманих результатів з експерименту, які розділимо на дві мережеві частини: внутрідоменну та позадоменну комунікації, з введеним протоколом для віддаленого доступу до пам'яті та механізмом спрощеної комунікації подвійно сумісним з працюючими на віртуальних машинах мережевими додатками. У ході експерименту було використано дві однакові фізичні машини, кожна з яких була оснащена двох-ядерним процесором з частотою 2,5 ГГц, об'ємом пам'яті 4 ГБ, SATA-диском, мережевою платою Gigabyte для комунікації в Gigabyte Ethernet мережі. Програмне забезпечення CentOS 5.0 з ядром Linux 2.6.18.8 і версією Xen 3.2.0 встановлювалося однаковим на обох фізичних машинах. Кожна робоча віртуальна машина була оснащена 2 VCPU з оперативною пам'яттю 512 Мбайт.

Експериментальна перевірка продуктивності для частини мережі, яка представляла позадоменний зв'язок, здійснювалася між двома фізичними машинами. Оцінка продуктивності частини мережі для внутрідоменного зв'язку виконувалася на одній фізичній машині, на якій згідно моделі були встановлені дві віртуальні машини, які, згідно системної моделі, відображали схематично сам домен та внутрішній зв'язок в ньому. Отримання кількісних результатів та оцінка параметрів продуктивності двох вищезгаданих мережевих ділянок здійснювалася за допомогою методу паралельного набору *NAS*-міток і *NETPERF*-міток, згідно методики, описаної в роботі [24]. Метод тестування з *паралельним набором NAS-міток* базується на наборі часових міток [25, 26], які відтворюють такі пристрої, як процесор, кеш-пам'ять, оперативна пам'ять і, звичайно, робочі пристрої вводу-виводу паралельно в часі у відповідному наборі працюючих додатків. Відомо [24], що

блок для отримання експериментальних даних методом *NAS*-міток з паралельним набором об'єднує в собі п'ять ядер EP, MG, CG, FT, IS та трьох динамічно працюючих додатків BT, SP, LU. Для проведення тестів згідно вказаної методики була вибрана проблема в розмірі A-класу. Слід також додати, що модуль для *NAS*-міток NPB3.3-MPI був розроблений і скомпільований в середовищі OpenMPI-l.4.2 на базі системи GCC 4.1.2 [24].

Графік залежності величини відносного робочого часу з використанням протоколу спрощеного зв'язку для отриманих наборів міток на основі п'яти вищеописаних діючих ядер та ядра BT зображено на рисунку 8. Порівнюючи отримані результати можна чітко помітити, що для діючого протоколу він є значно менший, ніж для TCP/IP протоколу. В ході виконання дослідження для отримання більшої достовірності експериментальних результатів був залучений тест з набором *Netperf*-міток. Одна з робочих віртуальних машин, яка знаходилася на фізичній машині, виконувала роль *Netperf*-сервера, а друга віртуальна машина, що знаходилася на іншій фізичній машині, повинна була в якнайшвидшому режимі організувати передачу пакетів великих обсягів на віртуальну машину-сервер.

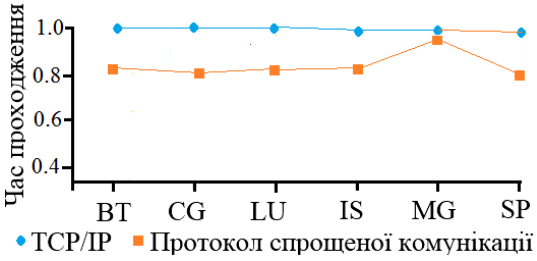

Рис. 8 – Залежність величини часу проходження від типу ядра для тесту з паралельним набором NAS-міток для класу A

В наступному було виконано тестування зі змінним часовим інтервалом вимірювання, як подано в роботі [27]. Отримані тестові результати (рисунок 9) констатують, що пропускна здатність даної мережевої ділянки з використанням протоколу спрощеного зв'язку, яка є оберненою величиною часу проходження і помітно вищою, порівняно із TCP/IP протоколом. Хоча вважається, що традиційний протокол TCP/IP відносно добре виконується в цьому середовищі і не вимагає його оптимізації роботи на якихось визначених конкретних мережевих ділянках, де б активізувався час затримки пакетів. Пропускна здатність зростає в основному на 2,11 %. Таке незначне покращення відбувається внаслідок скорочення часу запуску пакетів повідомлень та зменшення заголовкового пакета за розміром.

Для дослідження було задіяно декілька робочих віртуальних машин, що знаходилися на одній і тій же фізичній машині. Після їх старту в домені 0 був завантажений модуль відокремлення повідомлень, а на кожну робочу віртуальну машину завантажувався модуль XenLoop. Здійснимо порівняння пропускної здатності для традиційного протоколу TCP між двома робочими віртуальними машинами, що використовують Netperf між вхідним і вихідним драйверами механізму зв'язку та XenLoop-каналом. Далі перевіримо пропускну здатність для локального каналу замикання *loopback* та виконаємо порівняння з пропускною здатністю для каналу XenLoop.

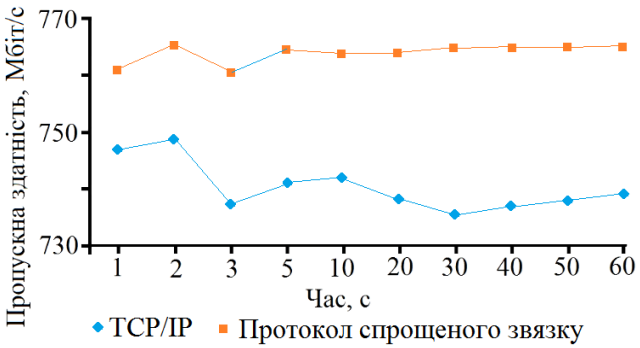

Рис. 9 – Результати зміни пропускної здатності для протоколів ПСЗ і TCP/IP в ході зміни часового інтервалу вимірювання

Рисунок 10 демонструє нам результат, який очевидно доводить, що для здійснення передачі великих пакетів даних пропускна здатність з використанням TCP-протоколу наближено покращується на півтора. Отже, пропускна здатність між взаємодіючими віртуальними машинами всередині домена, тобто співрезидентами на фізичній машині, покращується порівняно з пропускною здатністю для локального *loopback*-каналу.

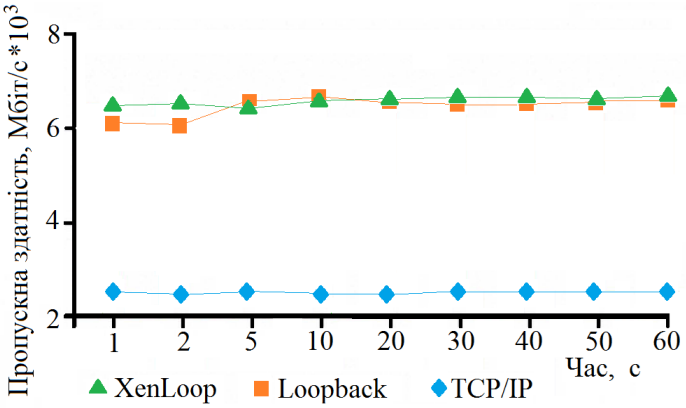

Рис. 10 – Результати тестування пропускної здатності для TCP/IP, XenLoop та Loopback

Виконаємо також порівняння отриманих даних з експерименту для запропонованої в роботі системи з протоколом спрощеного зв'язку і системи з діючим протоколом віддаленого доступу до пам'яті, які в дослідженні були використані окремо. Через них передавалася однакова визначена кількість пакетів з одної робочої віртуальної машини до іншої віртуальної машини, розміщених на різних фізичних машинах, які реально представляли позадоменний мережевий зв'язок. Результати, наведені на рисунку 11 підтверджують, що пропускна здатність, яка обернено пропорційна часу затримки, значно покращується у системі з використанням протоколу спрощеного зв'язку. Порівнюючи даний результат результатом, отриманим таким же підходом для системи з використанням протоколу віддаленого доступу до пам'яті, зростання приблизно спостерігається на 7,86–7,92 % в залежності від об'єму пакетів передавання. Не зважаючи на те, що протокол спрощеного зв'язку та протокол віддаленого доступу до пам'яті здійснюють відправку стільки даних в кожному пакеті, скільки максимально вони спроможні, все ж між ними відчувається суттєва різниця в пропускній здатності. Хоча протокол віддаленого доступу до пам'яті призначений для організації передавання сторінок пам'яті почергово (одна за одною), однак заключний пакет даних з розміром максимальної сторінкової пам'яті, призначений для мережевої передачі, не досягає точки призначення, тобто кінцевої віртуальної машини-одержувача. Отже, для протоколу віддаленого доступу до пам'яті порівняно з протоколом спрощеного зв'язку потрібно надіслати на кілька сторінок більше для пересилання наперед визначеної кількості інформації. Це говорить про те, що у випадку використання системи з механізмом спрощеного зв'язку та його протоколом, результуюча пропускна здатність значно покращується.

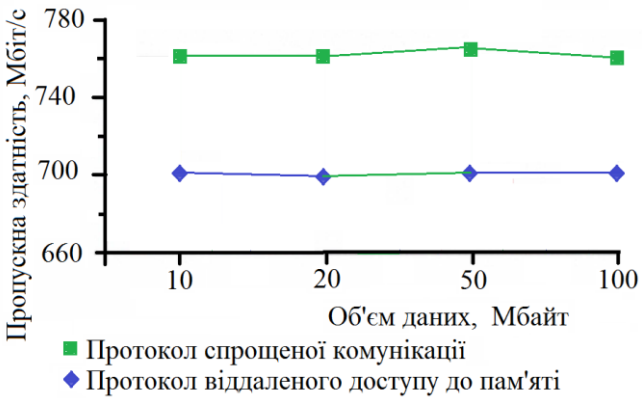

Рис. 11 – Залежність пропускної здатності від об'єму пакетів передавання для систем з протоколом спрощеного зв'язку та протоколом віддаленого доступу до пам'яті

Для перевірки на бінарну сумісність в ході роботи з різними додатками в системі з механізмом спрощеного зв'язку, потрібно було об'єднати в комунікації звичайні мережеві програми запущені на виконання. Для перевірки бінарної сумісності були запущені SCP, WGET, APACHE, MYSQL, NETPERF, TELNET, VSFTPD та інші додатки, які успішно були протестовані, за винятком тих, які не використовують традиційний інтерфейс для стандартних сокетів. Для мережевих додатків, у яких використовується стандартний сокетний інтерфейс, пропонована система з механізмом спрощеного зв'язку повністю підтримує бінарну сумісність в роботі з ними.

7. Висновки

В роботі представлено результати досліджень покращення внутрішніх та зовнішніх параметрів швидкодії зв'язку на кластері комунікуючих віртуальних машин для високопродуктивних обчислень. Для поліпшення внутрішньої та зовнішньої комунікації було застосовано систему з мережевим механізмом спрощеної комунікації, призначеним для підвищення продуктивності передачі пакетів у кластері. За рахунок введеного в систему спрощеного механізму зв'язку значно поліпшується пропускна здатність для внутрідоменної та позадоменної ділянок мережі в кластері, підтримуючи сумісність з одночасно працюючими мережевими додатками, які використовують звичайний сокетний інтерфейс, та уникаючи сповільнення, характерного для початкового етапу TCP-протоколу. Даний механізм дозволяє також використовувати менший заголовок для кожного пакета передавання, що також стимулює підвищення мережевої швидкодії та продуктивності.

Виявлено, що пропускна здатність між двома комунікуючими віртуальними машинами з використанням протоколу спрощеного зв'язку зростає приблизно на 2,11 %, ніж для системи таких же віртуальних машин з використанням традиційного TCP/IP протоколу. Поліпшення параметрів внутрідоменного зв'язку в системі зі спрощеною комунікацією базується на введенні в неї петлі XenLoop, за рахунок якої виконувалося розпізнавання та перехоплення надісланих пакетів на рівні нижче мережевого і перевірка призначення його будь-якій із співрезидентних машин, що утворювали внутрідоменний зв'язок. Пропускна здатність для мережі всередині домена покращується в середньому на півтора. Порівняння отриманих показників для системи з використанням протоколу спрощеного зв'язку клнмтатує відносне підвищення швидкодії передачі повідомлень на 7,8–7,9 %.

Список бібліографічного опису

1. S. Iannucci, V. Cardellini, O. D. Barba, I. Banicescu. (2020) A hybrid model-free approach for the near-optimal intrusion response control of non-stationary systems. // Future Generation Computer Systems. – 2020, V 109, p. 111-124.
2. D. Black, P. Jones (2015) Differentiated Services and Real-Time Communication, Informational. RFC 7657, – Nov. 2015. – IETF, ISSN 2070-1721.
3. D. Black, P. Jones, Differentiated Services and Real-Time Communication, Informational RFC 7657, IETF, ISSN 2070-1721 (Nov. 2015).
4. R. Barik,ˏM.Welzlˏ A. Elmokashfi, T. Dreibholzˏ S. Islamˏ S. Gjessing. (2019) On the utility of unregulated IP DiffServ Code Point (DSCP) usage by end systems. Performance Evaluation. – 2019, V. 135. – p. 102036.
5. Мельник, В.М. Побудова та використання міждоменного механізму зв'язку для високопродуктивної обробки даних [Текст] / В.М. Мельник, П.А. Пех, К.В. Мельник, Н.В. Багнюк, О.К. Жигаревич // Східно-європейський журнал передових технологій. – 2016. – № 1/9/79. – с. 10-15. doi: 10.15587/1729-4061.2016.60629.
6. Мельник, В.М. Вплив високопродуктивних сокетів на інтенсивність обробки даних [Текст] / Мельник В.М., Багнюк Н.В., Мельник К.В. // Scientific Journal "ScienceRise". – 2015. – Том 6, № 2(11). – с. 38-48. doi: 10.15587/2313-8416.2015.44380.
7. Melnyk, V. High production of java sockets (HPJS) for health clouds in science [Text] / V. Melnyk, K. Melnyk, O. Zhyharevych // Інженерія програмного забезпечення. Національний авіаційний університет. – 2015. – Том 19, № 3. – С. 36-40.
8. Melnyk, V. Significance of the socket programming for the laboratory with intensive data communications [Text] / V.Melnyk, P.Pekh, K.Melnyk, O. Zhyharevych // Всеукраїнська Науково-практична інтернет-конференція «Сучасні методи, інформаційне та програмне забезпечення систем управління організаційно-технологічними комплексами». (28 квітня 2015 р.). – Луцьк, Луцький НТУ. / Науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво.» – 2015, №20. – с. 67-71.
9. Бархем, П. Xen і мистецтво віртуалізації [Текст] / П. Бархем, Б. Драгович, К.Фрейзер та ін. // Праці симпозіуму ACM на операційних системах принципів (SOSP 2003), ACM Press, Нью-Йорк, США: Болтон Landing, Lake George, жовтень 2003. – с. 164-177.
10. Пратт, І. Xen-Віртуалізація. Linux в світі, 2005 [Текст] / І. Пратт // Віртуалізація BOF-презентація. 2007.
11. Chisnall, D. Повне керівництво по Xen Hypervisor [Текст] / D. Chisnall // Prentice Hall, 2-е видання, 2007.
12. Менон, А. Оптимізація віртуалізації мережі в Xen. У 2006 р [Текст] / А. Менон, А. Л. Кокс, В. Звейніпоель // USENIX щорічна технічна конференція, Бостон, штат Массачусетс, США, червень 2006. – с. 15-28.
13. Jian, W. XenLoop: Прозора висока продуктивність між віртуальними машинами через мережевий шлейф [Текст] / В.Цзянь, К. L.Райт, К.Гопалан // Бостон, штат Массачусетс, США, HPDC'08, 23-27 червня 2008.
14. Кім, К. Міжсокетний зв'язок і підтримка високої продуктивності і повної бінарної сумісності на Xen [Текст] / К. Кім, С. Кім, С.-І. Юнг, Г.-С. Шін, Дж.-С. Кім // Сіетл Вашингтон, США, VEE'08, 5-7 березня 2008.
15. Закон Амдаля. Електронний ресурс. Режим доступу: http://home.wlu.edu/~whaleyt/classes/parallel/topics/amdahl.html.
16. V. Melnyk, N. Bahnyuk, K. Melnyk, O. Zhyharevych, N. Panasyuk. Implementation of the simplified communication mechanism in the cloud of high performance computations. East-European journal of Enterprise Technologies. – Kharkiv (DOI: 10.15587/1729-4061.2017.98896). – 2017. – № 2/2/86. – p. 24–32.

17. Хуан, Дж. Л. В. Висока продуктивність обходу між MBM для вводу/виводу в віртуальних машинах [Текст] / Дж. Л. В. Хуан, Б. Абалі, Д. К. Панда // USENIX щорічна технічна конференція Архів, 2006.
18. Хайнс, М. Р. MemX: Підтримка великих робочих навантажень пам'яті в Xen для віртуальних машин [Текст] / М. Р. Хайнс, К. Гопалан // VTDC '07, Рено, Невада, США. 12 листопада, 2007.
19. Дешпанде, У. Memx: Віртуалізація кластера з широкою пам'яттю [Текст] / Дешпанде У., Ван Б., Хак С., Хайнс М., К. Ґопалан // У ICPP'IO: Праці 39-й Міжнародної конференції з паралельної обробки (2010), С. 663–672.
20. Кім, Й.-С. Проектування та реалізація рівня користувача та рівня сокета над віртуальною архітектурою інтерфейсу [Текст] / Й.-С. Кім, К. Кім, С.-І. Юнг, С. Ха // Паралельне Обч.: Практ. Експер, 15 (7-8). – 2003. – с. 727–749.
21. Син, С. ВОП: сокет-інтерфейс для ВО [Текст] / С. Син, Д. Кім, Є. Лім, С. Юнг // В Інтернеті та мультимедійних системах і додатках, 2004.
22. Кларк, Д. Д. Вікно і стратегія визнання TCP [Текст] / Д. Д. Кларк // RFC 813, Цільова група Інтернет інженерії. 1982. Електронний ресурс. Режим доступу: https://tools.ietf.org/html/rfc813.
23. Menon A. Diagnosing Performance Overheads in the Xen Virtual Machine Environment [Текст] / A. Menon , J.R. Santos, Y. Turner, J. Janakiraman, W. Zwaenepoel. – May 3, 2005. Електронний ресурс. Режим доступу: https://www.hpl.hp.com/techreports/2005/HPL-2005-80.pdf.
24. Семёнушкин А.В. Тестирование пропускной способности сети. Електронний ресурс. Режим доступу: /http://semenushkin.ru/2010/07/01/тестирование-пропускной-способности/.
25. Бейлі, Д.Г. "Паралельні тести NAS [Текст] / Д.Г. Бейлі, Е. Баржш, Дж. Бартон, Д. Браунінг, Р. Картер, Д. Дагум, П. А. Фатоугі, П. О. Фредріксон, Т. Ласінскі, Р. С. Шрайбер, Н. Д. Симон, Б. Венкатакрішнан, С. К. Віратунга //" Міжнародний журнал суперкомп'ютерних додатків. – (осінь +1991), т. 5, № 3, С. 63-73.
26. Обзор некоторых пакетов измерения производительности кластерных систем. Електронний ресурс. Режим доступу: http://www.ixbt.com/cpu/cluster-benchtheory.shtml.
27. Netperf: мітки мережевої продуктивності. Редакція 2.0./ Компанія Гевлет-Пекерд. Електронний ресурс. Режим доступу: http://www.netperf.org/netperf/training/Netperf.html.

References

1. S. Iannucci, V. Cardellini, O. D. Barba, I. Banicescu. (2020) A hybrid model-free approach for the near-optimal intrusion response control of non-stationary systems. // Future Generation Computer Systems. – 2020, V 109, p. 111-124.
2. D. Black, P. Jones (2015) Differentiated Services and Real-Time Communication, Informational. RFC 7657, – Nov. 2015. – IETF, ISSN 2070-1721.
3. D. Black, P. Jones, Differentiated Services and Real-Time Communication, Informational RFC 7657, IETF, ISSN 2070-1721 (Nov. 2015).
4. R. Barik,'M.Welzl A. Elmokashfi, T. Dreibholz S. Islam S. Gjessing (2019) On the utility of unregulated IP DiffServ Code Point (DSCP) usage by end systems. Performance Evaluation. – 2019, V. 135. – p. 102036.
5. Melnyk, V., Pekh, P., Melnyk, K., Bahnyuk, N., Zhyharevych O. (2016) The cross-domain communication mechanism construction and use for high-performance data computing. The East-European Journal of Advanced Technologies. Kharkiv. 1/9/79, 10-15. DOI: 10.15587/1729-4061.2016.60629.
6. Melnyk, V., Bahnyuk, N., Melnyk, K. (2015). The influance of high-performance sockets on the data computing intensity. Scientific Journal ScienceRise, Kharkiv, 6, 2(11), 38-48. DOI: 10.15587/2313-8416.2015.44380.
7. Melnyk, V. High production of java sockets (HPJS) for health clouds in science [Text] / V. Melnyk, K. Melnyk, O. Zhyharevych // Software engineering. National Aviation University. – 2015. – Vol. 19, № 3. – p. 36–40.
8. Melnyk, V. Significance of the socket programming for the laboratory with intensive data communications [Text] / V.Melnyk, P.Pekh, K.Melnyk, O. Zhyharevych // Ukrainian Scientific and Practical Internet Conference "Modern Methods, Information and Software for Management Systems of Organizational and Technological Complexes". (April 28, 2015). Lutsk, Lutsk NTU. / Scientific journal "Computer-integrated technologies: education, science, production." - 2015, №20. – p. 67–71.
9. Barham, P. Xen and the art of virtualization [Text] / P. Barham, B. Dragovich, K. Fraser and others. // Proceedings of the ACM Symposium on Operating Systems Principles (SOSP 2003), ACM Press, New York, USA: Bolton Landing, Lake George, October 2003. – p. 164–177.
10. Pratt, I. (2007). Xen Virtualization. Linux world 2005 Virtualization BOF Presentation.
11. Chisnall, David. (2007). The Definitive Guide to the Xen Hypervisor. Prentice Hall, 2-nd edition.
12. Menon, A., Cox, A. L., Zwaenepoel, W. (2006). Optimizing network virtualization in Xen. In 2006 USENIX Annual Technical Conference, Boston, Massachusetts, USA, 15–28.
13. Jian, W. XenLoop: Transparent high performance between virtual machines through a network loop [Text] / W. Jian, K. L. Wright, K. Gopalan // Boston, Massachusetts, USA, HPDC'08, 23–27 June 2008.
14. Kim, K., Kim, C., Jung, S.-I., Shin, H.-S., Kim, J.-S. (March 5–7, 2008). Inter-domain Socket Communications Supporting High Performance and Full Binary Compatibility on Xen.VEE'08, Seattle, Washington, USA.
15. Amdal's law. Electronic resource. Access mode: http://home.wlu.edu/~whaleyt/classes/parallel/topics/amdahl.html.
16. V. Melnyk, N. Bahnyuk, K. Melnyk, O. Zhyharevych, N. Panasyuk. Implementation of the simplified communication mechanism in the cloud of high performance computations. East-European journal of Enterprise Technologies. – Kharkiv (DOI: 10.15587/1729-4061.2017.98896). – 2017. – № 2/2/86. – p. 24–32.
17. Juan, J. L. V. High productivity of bypass between MVM for input / output in virtual machines [Text] / J. L. V. Juan, B. Abali, D. K. Panda // USENIX annual technical conference Archive , 2006.
18. Hines, MR MemX: Support for large memory workloads in Xen for virtual machines [Text] / MR Hines, K. Gopalan // VTDC '07, Renault, Nevada, USA. November 12, 2007
19. Deshpande W. Memx: Virtualization of a cluster with a wide memory [Text] / Deshpande W., Van B., Huck S., Hines M., K. Gopalan // In ICPP'IO: Proceedings of the 39th International Conference on Parallel Processing (2010), pp. 663–672.
20. Kim J.-S. Design and implementation of user level and socket level over virtual interface architecture [Text] / J.-S. Kim, K. Kim, S.-I. Jung, S. Ha // Parallel Obch .: Practice. Expert, 15 (7–8). – 2003. – p. 727–749.
21. Sin S. VOP: socket-interface for VO [Text] / S. Sin, D. Kim, E. Lim, S. Jung // In the Internet and multimedia systems and applications, 2004.
22. Clark D.D. Window and TCP recognition strategy [Text] / DD Clark // RFC 813, Target group of Internet engineering. 1982

23. Menon A. Diagnosing Performance Overheads in the Xen Virtual Machine Environment [Текст] / A. Menon , J.R. Santos, Y. Turner, J. Janakiraman, W. Zwaenepoel. – May 3, 2005. Electron resource. Access mode: https://www.hpl.hp.com/techreports/2005/HPL-2005-80.pdf.
24. Semenushkin A. Network bandwidth testing. Electronic resource. Access mode: /http://semenushkin.ru/2010/07/01/testirovanie-propusknoy-posobnosti/.
25. Bailey, D.G. "Parallel NAS tests [Text] / DG Bailey, E. Barges, J. Barton, D. Browning, R. Carter, D. Dagum, PA Fatougi, P. O. Fredrickson, T. Lasinski, RS Schreiber, ND Simon, B. Venkatakrishnan, SK Viratunga // "International Journal of Supercomputer Applications. - (autumn +1991), vol. 5, № 3, pp. 63-73.
26. Overview of some cluster performance measurement packages. Electronic resource. Access mode: http://www.ixbt.com/cpu/cluster-benchtheory.shtml.
27. Netperf: network performance labels. Edition 2.0./ Hewlett-Packerd Company. Electronic resource. Access mode: http://www.netperf.org/netperf/training/Netperf.html.