Интегриране на Системата за големи данни със Суперкомпютъра Петаскейл в ТехПарк София

Съдържание

1. Въведение 2

2. Функционални възможности на Системата за големи данни, предоставяни на Суперкомпютър 3

3. Функционални възможности на Суперкомпютър, предоставяни на Системата за големи данни 4

4. Изграждане на едноточково свързване на Комутатора на Много-клъстерната Hadoop Система за големи данни в УНСС с Комутатора в ТехПарк София, на базата на MAN мрежа 5

4.1. Активизиране на физическата свързаност 6

4.2. Наличието на двупосочни MAC адреси 6

4.3. Тестване на скоростта за предаване на данни по изградената MAN мрежа 7

5. Представяне на данните в Системите за големи данни като NFS файлове за ползване от Суперкомпютър 9

6. Стартиране на приложение в Суперкомпютър от Система за големи данни 12

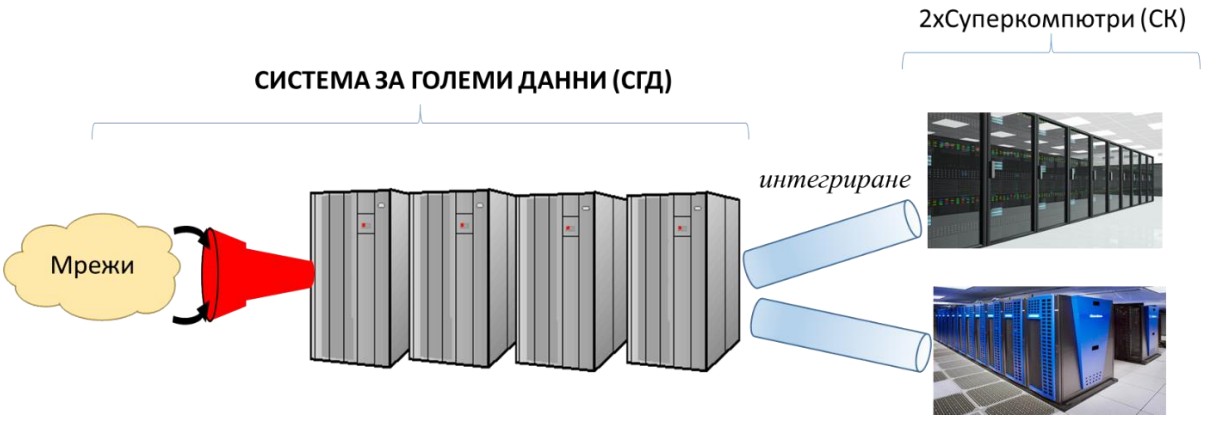
Литература 16

1. Въведение

Системата за големи данни Hadoop разположена в УНСС София (СГД) е изградена са основата на Apache Hadoop под управление на Cloudera управлението. Архитектурата й се състои от 3 броя NameNode, 2 броя ManagementNode, EdgeNode и множество DataNode. Всеки от DataNode има 128GB RAM памет (оперативна памет) и 120TB дискова памет, като общото дисково пространство на Системата за големи данни е около 4,5 PB, а общата RAM памет е около 4,6TB, а общият брой ядра в Системата за големи данни е 576. Специфичното на архитектурата на Системата за големи данни е че е раково ориентирана – DataNode-овете са разположени в отделни ракове, като мрежовата връзка (локалната мрежа) между отделните ракове е със скорост 40Gbps, а мрежовата връзка между отделните DataNode е със скорост 10Gbps. Всеки DataNode е изграден от процесори с архитектура CPU Intel Xeon Bronze. Системата за големи данни е защитена от UPS система, подсигурена от дизел генератор.

Discoverer петаскейл Суперкомпютърът в Тех Парк София има размер, съчетаващ 12 изчислителни Direct Liquid Cooling BullSequana сървърни рака. Платформата на Суперкомпютъра е изградена върху AMD EPYC процесори, охлаждани с вода, изграждащи 376 изчислителни възела. Броят на процесорните ядра е 144,384 със сумарна оперативна памет от 300TB. Оптималната функционалност е гарантирана от 2 PB висикоскоростно дисково пространство за съхранение на данни. Суперкомпютърът е защитен от прекъсване на електричеството чрез непрекъсваемо електрическо захранване с мощност от 1 MW.

Системата за големи данни Hadoop е свързана с 10Gbps MAN връзки с основни градове в България. Осигурени са също така и 2 броя самостоятелни връзки от по 10Gbps със Суперкомпютъра в Тех Парка и със предстоящия за пускане в експлоатация друг суперкомпютър в ИИКТ на БАН - фиг.1



Фиг.1

Системата за големи данни в УНСС е свързана с мрежи за комплексно приемане и изпращане на данни. Тези мрежи са следните 5 броя:

- MAN с общ капаците на предавани данни 40Gbps и с преки връзки с Пловдив, Габрово, Русе и Варна с по 10Gbps всяка;
- WAN с 10Gbps;
- Национална LoRaWAN мрежа, покриваща основните градове в България;
- TTN – Международна LoRaWAN мрежа осигуряваща предаване на данни с над 110 държави по света;
- 4G/5G мрежа за устройства разположени в тези мрежи.

2. Функционални възможности на Системата за големи данни, предоставяни на Суперкомпютър

При интегрирането на Системата за големи данни Hadoop на УНСС със Суперкомпютър, Системата за големи данни може да изпълнява следните функции:

a) Транзитно предаване на данни за Суперкомпютър. Това означава, използвайки множеството мрежови системи с висока надеждност на СГД, да приема данни от различни източници и след потенциална предварителна обработка, да ги подава директно на Суперкомпютър за обработка;

b) Съхранение на данни на Суперкомпютър. Това означава
   - да приема данни от различни източници и да ги съхранява в свое дисково пространство, за последващо използване от Суперкомпютър – съхранение на данни на Суперкомпютър,
   - да приема определи полу-структурирани и пълно не-структурирани данни за Преподготовка и да ги съхранява за последваща обработка от Суперкомпютър

- да съхранява резултатни данни от обработка на Суперкомпютъра с оглед последващ Анализ, последващо предаване от мрежите на СГД на Краен потребител, или просто за съхранение поради липса на достатъчно дисково пространство в Суперкомпютъра.

c) Да извършва Анализ на получени резултати от работа на Суперкомпютъра, поради наличието на множество софтуерни средства за анализ на данни в Системите за големи данни.
d) Суперкомпютърът да получи като резултат частично формирани дзанни, които да се предадат на Системата за големи данни, за последваща обработка чрез Impala, Spark или потребителски създаден програмен код.
e) Да представлява Обект на киберсигурност на данни за Суперкомпютър;
f) Да изпълнява ролята на Инструмент за киберсигурност на данни за Суперкомпютър;

3. Функционални възможности на Суперкомпютър, предоставяни на Системата за големи данни

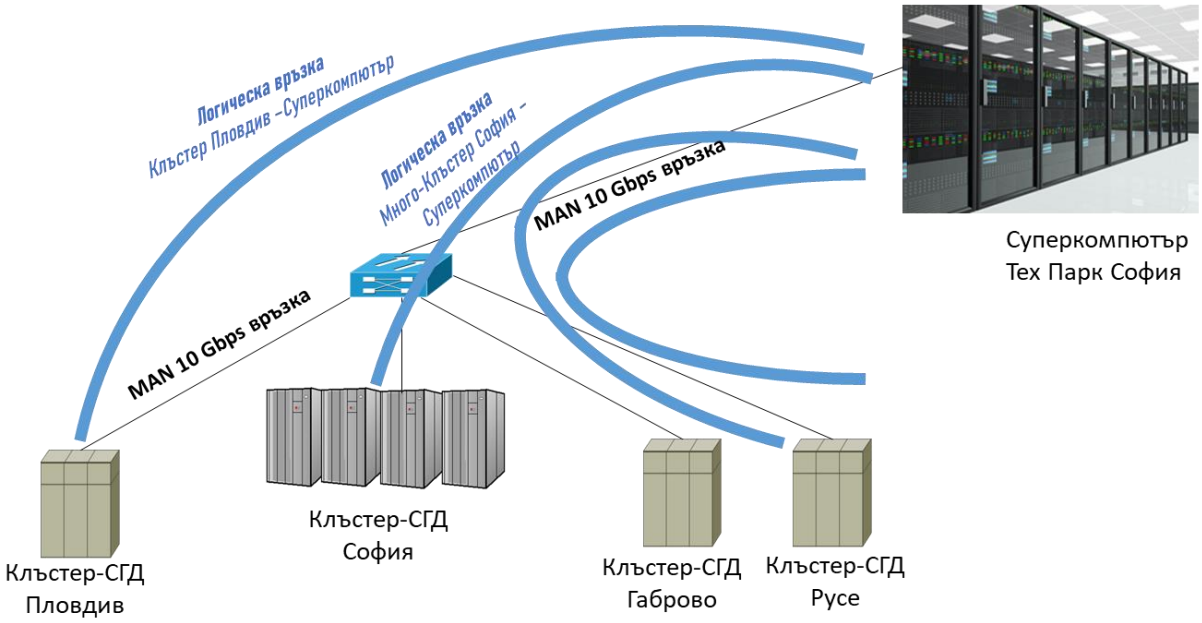
Суперкомпютърът може да се разглежда като допълнителен сървър към Системата за големи данни, който да извършва специализирани високо производителни изчисления. Естествено архитектурно разширение на Системата за големи данни е да работи съвместно с SQL Бази от данни (SQL базирани Релационни бази от данни), с ERP (Enterprise Resource Planning) системи, със Системи за управление на бизнес процеси (BPM – Business Process Management), със системи за Управление на съдържанието (CMS – Content Management System), или с NoSQL Бази от данни, които не се интегрират в Системите за големи данни.

За да може Суперкомпютърът да се използва като специализиран сървър за високопроизводителни изчисления на Ситемита за големи данни, то

4. Изграждане на едноточково свързване на Комутатора на Много-клъстерната Hadoop Система за големи данни в УНСС с Комутатора в ТехПарк София, на базата на MAN мрежа

Изградена е дедикирана еднопосочна връзка (точена-в-точка) между Системата за големи данни Hadoop в УНСС, която по същество е много-клъстерна Hadoop система (състояща се от 4 Hadoop клъстера свързани в много-клъстерна система) с MAN комутатор в Тех Парк София, към който е свързан Суперкомпютъра. Много-клъстерната Hadoop система се състои от Централен клъстер в УНСС, от клъстер в Пловдивския университет „Паисий Хилендарски“ , Клъстер в Техническия университет - Габрово и от Клъстер в Русенския университет „Ангел Кънчев“.

Концептуалната Архитектура на създадената много-клъстерна връзка със Суперкомпютъра е представена на фигура 2.



Фиг.2

Физически, всичките посочени 4 броя Клъстери на СГД са свързани с Комутатор, разположен в УНСС – София, който от своя страна е свързан със Суперкомпютъра в Тех Парка София. Това означава, че всеки от посочените 4 броя Hadoop клъстера са свързани логически със Суперкомпютъра, т.е. осъществени са следните Логически връзки между Hadoop клъстер и суперкомпютър:

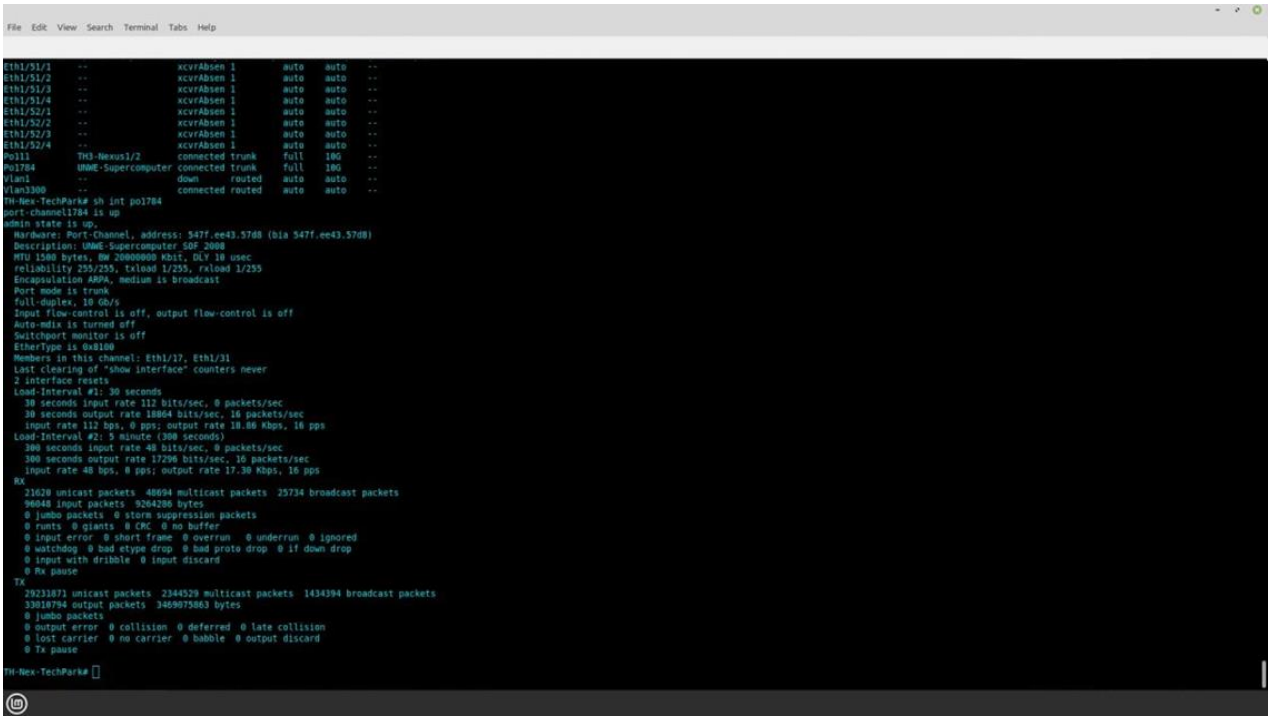
- Клъстер-СГД в УНСС – София със Суперкомпютър в Тех Парка София;

- Клъстер-СГД в Пловдивския университет „Паисий Хилендарски“ – Пловдив със Суперкомпютър в Тех Парка София;
- Клъстер-СГД в Техническия университет - Габрово със Суперкомпютър в Тех Парка София;
- Клъстер-СГД в Русенския университет „Ангел Кънчев“ – Русе със Суперкомпютър в Тех Парка София.

Това означава, че Много-клъстерната Hadoop система осигурява интегриране на 4 университета в България със Суперкомпютъра в Тех Парка София - УНСС, Пловдивския университет „Паисий Хилендарски“ , Техническия университет - Габрово и Русенския университет „Ангел Кънчев“. След изграждането на MAN връзките между 4-те посочени университети, се изгражда една връзка между УНСС и Тех Парк София. Създаването на тази връзка и нейното тестване е представено по-долу:

4.1. Активизиране на физическата свързаност

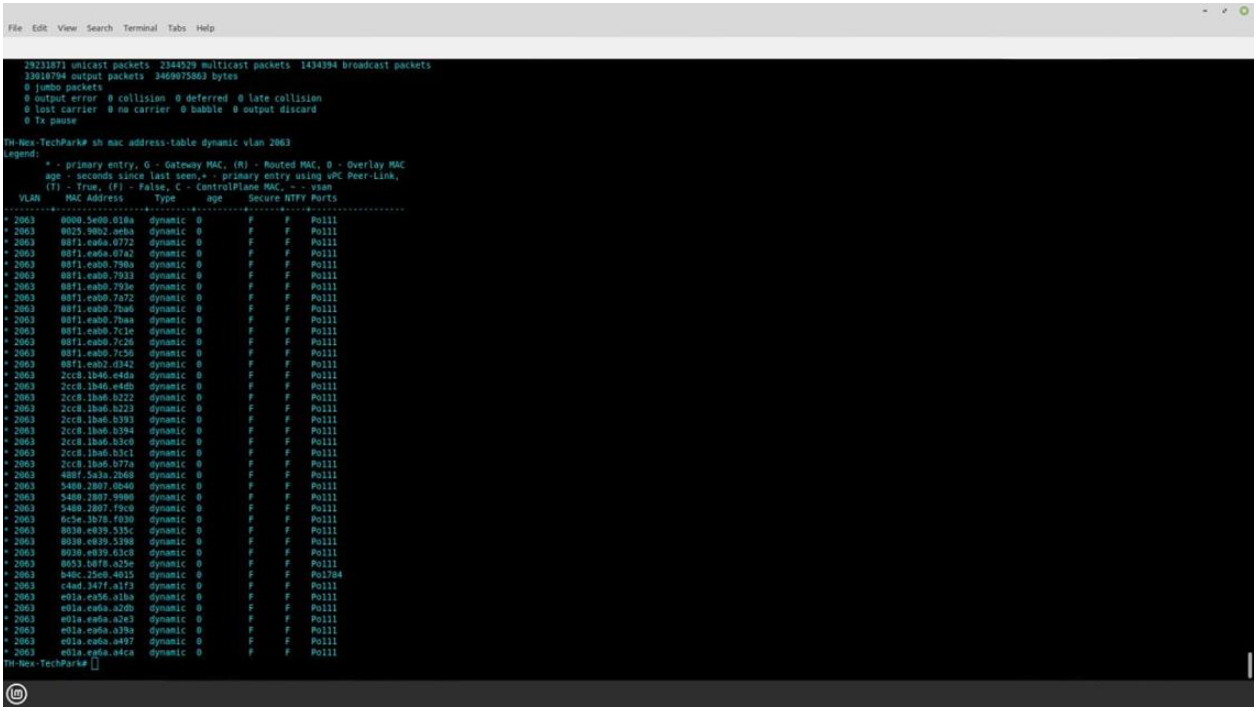
На фигура 3 е показана активността на физическата свързаност (статус на порта „UP“).



Фиг.3

4.2. Наличието на двупосочни MAC адреси

На фигура 4 е показана представянето на двупосочни MAC адреси на двете части на физическата свързаност.



Фиг.4

4.3. Тестване на скоростта за предаване на данни по изградената MAN мрежа

За да се тества скоростта за предаване на данни по изградената MAN мрежа, се използва софтуерният инструмент iperf3.

Инструментът perf е инструмент за измерване и настройка на производителността на мрежата. Това е междуплатформен инструмент, който може да произвежда стандартизирани измервания на производителността за всяка мрежа. Iperf има функционалност за клиент и сървър и може да създава потоци от данни за измерване на пропускателната способност между двата края в едната или в двете посоки. Типичният изход на iperf съдържа отчет с щамповано време за количеството прехвърлени данни и измерената пропускателна способност.

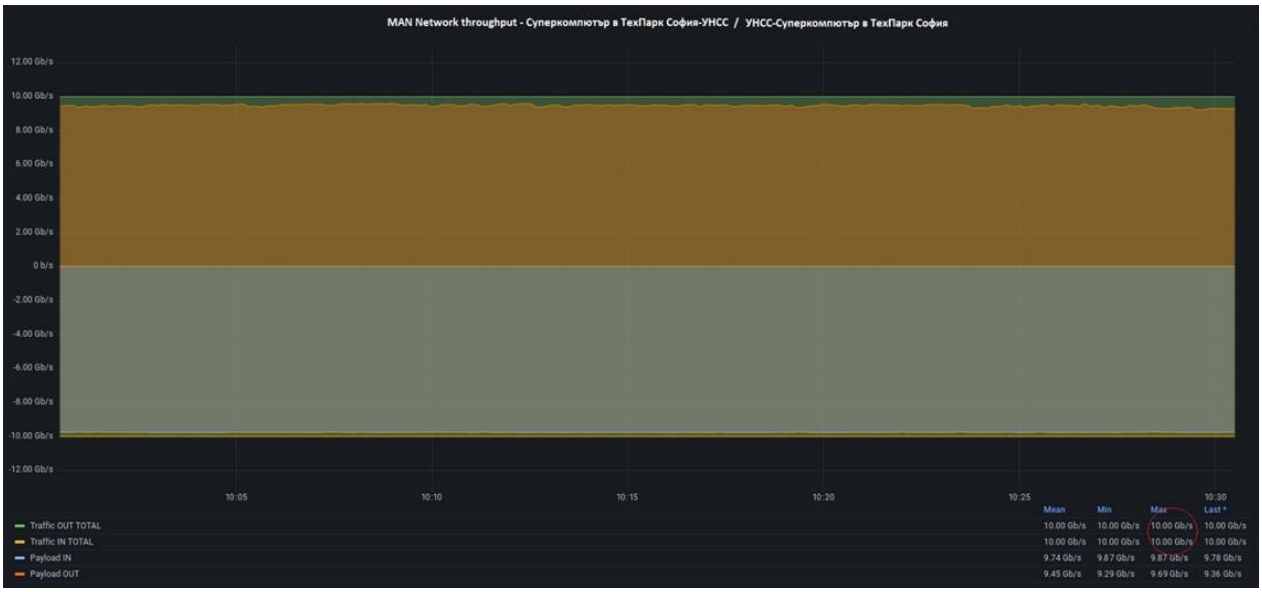
Потоците от данни могат да бъдат или протокол за управление на предаването (TCP), или протокол за потребителска дейтаграма (UDP):

- UDP: Когато се използва за тестване на UDP капацитет, iperf позволява на потребителя да посочи размера на дейтаграмата и предоставя резултати за пропускателната способност на дейтаграмата и загубата на пакети.
- TCP: Когато се използва за тестване на TCP капацитет, iperf измерва пропускателната способност на полезния товар. Iperf използва 1024 × 1024 за мебибайти и 1000 × 1000 за мегабайти.

Iperf е софтуер с отворен код, написан на C, и работи на различни платформи, включително Linux, Unix и Windows. Наличието на изходния код позволява на потребителя да разгледа внимателно методологията за измерване.

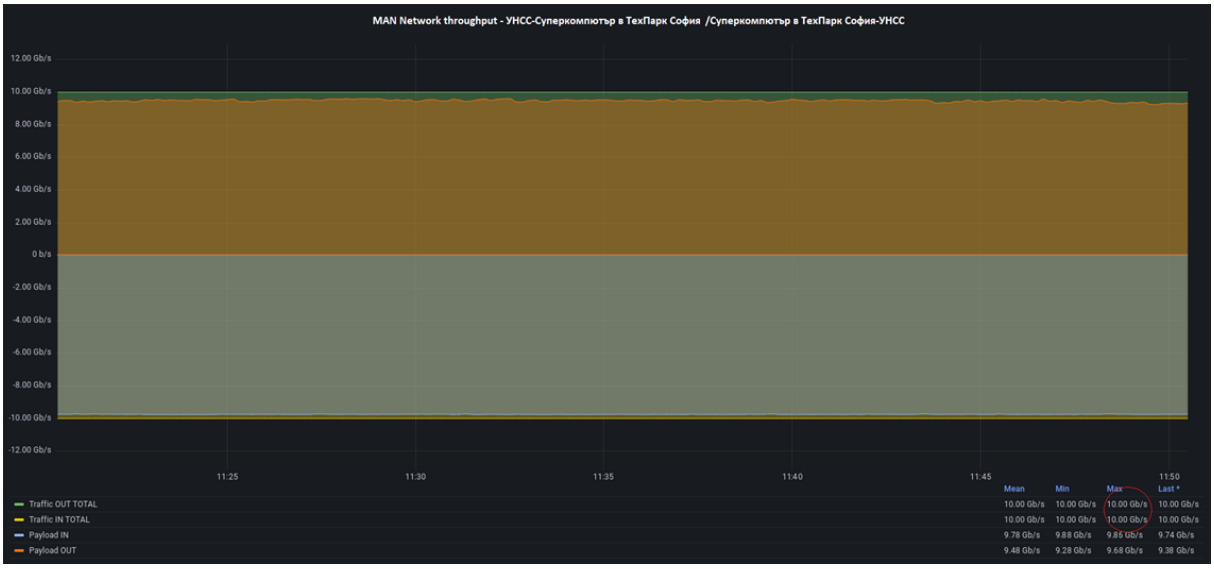
Iperf3 е пренаписване на iperf от нулата, за да се създаде по-малка, по-проста кодова база. Той също така включва версия на библиотека, която позволява на други програми да използват предоставената функционалност. Iperf3 е еднонишков, докато iperf2 е многонишков. Iperf3 стартира през 2009 г. с първото издание през януари 2014 г. Iperf3 не е обратно съвместим с iperf2. Iperf3 официално поддържа само Linux.

На фигура 5 е показана Разпечатка от iperf3 тест направен от Комутатора в Суперкомпютър в ТехПарк София към Комутатора на Много-клъстерната разпределена Hadoop система в УНСС двупосочно, доказващ скоростта на предаване на данни 10Gbps.



Фиг.5

На фигура 5 е представена Разпечатка от iperf3 тест направен от Комутатора на Много-клъстерната разпределена Hadoop система в УНСС към Комутатора в Суперкомпютър в ТехПарк София двупосочно, доказващ скоростта на предаване на данни 10Gbps.



Фиг.6

5. Представяне на данните в Системите за големи данни като NFS файлове за ползване от Суперкомпютър

Системата за големи данни, оперираща с HDFS файлове, предлага много сериозни предимства на Суперкомпютъра, оперираща с NFS файлове, защото HDFS файловете са с вградена защита от грешки (толерантност от грешки) за разлика от NFS файловата система. Системата за големи данни прави симулация на HDFS файловете като NFS файлове, за да могат да се използват от Суперкомпютъра, но същевременно да предоставя толерантност от греши.

По принцип, NFS софтуерната компонента в Системите за големи данни (наречена още NFS Gateway), позволява достъп до файловете, сякаш файловете се намират на локалната машина, въпреки че се намират Системата за големи данни като HDFS файлове. Основната разлика между NFS и HDFS файлови системи е репликация/толерантност към грешки. HDFS е проектиран да осигури продължителност на функциониране при откази. NFS няма вградена толерантност към грешки. Освен толерантността към грешки, HDFS поддържа множество реплики на файлове. Това елиминира (или облекчава) обичайното затруднение на много клиенти на Суперковпютър, които имат достъп до един файл. Тъй

като файловете имат множество реплики, на различни физически дискове разположени на различни физически ракове, производителността на четене се мащабира по-добре от NFS.

NFS Gateway в Системата за големи данни Hadoop поддържа NFSv3 и позволява HDFS данните като NFS данни да бъдат „монтирани“ към операционната система на Суперкомпютъра (команда за правене на достъпност от Суперкомпютъра до HDFS данните в Системата за Големи данни) и да се оперира с тях като част от локалната файлова система на Суперкомпютъра. В момента NFS Gateway поддържа и разрешава следните модели на използване:

- Потребителите на Суперкомпютъра могат да разглеждат HDFS файловата система през тяхната локална файлова система.
- Потребителите на Суперкомпютъра могат да изтеглят файлове от файловата система HDFS в тяхната локална файлова система.
- Потребителите на Суперкомпютъра могат да качват файлове от тяхната локална файлова система директно във файловата система HDFS.
- Потребителите на Суперкомпютъра могат да оперират директно с данни от HDFS. Чрез Суперкомпютъра може да се разширява файл и да се добавя файл, но не се поддържа произволно записване (поради спецификата на Системата за големи данни).

NFS Gateway функционира по същия начин както клиент като Hadoop JAR файлове. За целта NFS Gateway може да бъде инсталиран на същия хост (сървър в Hadoop системата) като DataNode или на NameNode.

NFS Gateway използва прокси потребител за прокси на всички потребители, които имат достъп до NFS монтирания. В незащитен режим потребител-администратор на Системата за големи данни, управляващ NFS Gateway, е прокси потребител, докато в защитен режим потребителят-администратор в Системата за големи данни в Kerberos keytab е прокси потребител. Да предположим, че прокси потребителят е „nfsserver“ и потребителите, принадлежащи към групите „users-group1“ и „users-group2“, използват NFS монтирания, тогава в core-site.xml на NameNode трябва да бъдат зададени следните две свойства и само NameNode трябва да се рестартира след промяната на конфигурацията. Подобно конфигуриране е следното:

```
<property>
  <name>hadoop.proxyuser.nfsserver.groups</name>
  <value>root,users-group1,users-group2</value>
  <description>
```

```
         The 'nfsserver' user is allowed to proxy all members of the
'users-group1' and
         'users-group2' groups. Note that in most cases you will need to
include the
         group "root" because the user "root" (which usually belonges to
"root" group) will
         generally be the user that initially executes the mount on the NFS
client system.
         Set this to '*' to allow nfsserver user to proxy any group.
  </description>
</property>

<property>
  <name>hadoop.proxyuser.nfsserver.hosts</name>
  <value>nfs-client-host1.com</value>
  <description>
         This is the host where the nfs gateway is running. Set this to '*'
to allow
         requests from any hosts to be proxied.
  </description>
</property>
```

Горната е единствената необходима конфигурация за NFS Gateway в незащитен режим. За Kerberized hadoop клъстери трябва да се добавят следните конфигурации към hdfs-site.xml за шлюза (ЗАБЕЛЕЖКА: заменете низа „nfsserver“ с прокси потребителското име и се уверете, че потребителят, съдържащ се в keytab, също е същият прокси потребител):

```
<property>
   <name>nfs.keytab.file</name>
   <value>/etc/hadoop/conf/nfsserver.keytab</value> <!-- path to the nfs gateway keytab -->
 </property>


 <property>
   <name>nfs.kerberos.principal</name>
   <value>nfsserver/_HOST@YOUR-REALM.COM</value>
 </property>
```

Ако се изисква да има достъп до HDFS NFS Gateway от UNIX-based операционна система, трябва да се зададе следната конфигурационна настройка:

```
<property>
  <name>nfs.aix.compatibility.mode.enabled</name>
  <value>true</value>
</property>
```

HDFS супер-потребител е потребител със права за достъп до NameNode и супер-потребителят може да направи всичко, така че проверките на разрешения никога да не се провалят за супер-потребителя. Ако е конфигурирано следното свойство, супер-потребителят на NFS клиент може да има достъп до всеки файл на HDFS. По подразбиране супер потребителят не е конфигуриран в NFS Gateway. Дори супер-потребителят да е конфигуриран, „nfs.exports.allowed.hosts“ все още влиза в сила. Например, супер-потребителят няма да има достъп за запис до HDFS файлове през шлюза, ако на NFS клиентския хост не е разрешен достъп за запис в „nfs.exports.allowed.hosts“

```
<property>
  <name>nfs.superuser</name>
  <value>the_name_of_hdfs_superuser</value>
</property>
```

6. Стартиране на приложение в Суперкомпютър от Система за големи данни

Планировчикът е софтуер, който внедрява за изпълнение пакетна система на Суперкомпютър. Потребителите на Суперкомпютър не изпълняват своите изчисления директно и интерактивно (както правят на личните си работни станции или лаптопи), вместо това те изпращат неинтерактивни пакетни задания към планировчика. Планировчикът съхранява пакетните задачи, оценява техните изисквания за ресурси и приоритети и разпределя задачите към подходящи изчислителни възли. За разлика от възлите за влизане (за компилиране и тестване на потребителски софтуер) и тяхното интерактивно използване, изчислителните възли в Суперкомпютъра обикновено не са директно достъпни (например чрез ssh). По този начин планировчикът е интерфейсът за потребителите на Суперкомпютрите за влизане, за да изпращат работа до изчислителните възли. Това изисква потребителят да попита планировчика за време и ресурси на паметта и да посочи приложението в скрипт на задание. След това този скрипт за задание може да бъде подаден към пакетната система чрез планировчика, който първо ще добави заданието към опашка за задания. Въз основа на ресурсите, от които заданието се нуждае, планировчикът ще реши кога заданието ще напусне опашката и на кои (част от) задните възли ще се изпълнява.

Най-общо казано, всеки планировчик има три основни цели:

- минимизиране времето между подаването на заданието и завършването на работата: нито едно задание не трябва да остава на опашката за дълги периоди от време
- оптимизиране на използването на процесорите на Суперкомпютъра: централните процесори на суперкомпютъра са един от основните ресурси за голямо приложение; следователно трябва да съществуват само няколко времеви интервала, в които процесорът не работи
- увеличаване максимално производителността на работата като се продават за изпълнение възможно най-много задачи за единица време в Суперкомпютъра.

SLURM (Simple Linux Utility for Resource Management) е система за управление на клъстери и планиране на задачи. Това е софтуерът, който най-много се използва в Суперкомпютрите за управление на ресурсите. Това е софтуер с отворен код.

За да се изпълни задача към Суперкомпютър, първо трябва да се осъществи свързване с възел за подаване на заявки. За всеки клъстер има поне един възел за подаване на заявки с име <cluster>-gw, напр.: phoenix-gw, hm-gw и т.н.

Основните функциите на Slurm включват:

- Силно мащабируемост (планира до 100 000 независими задания на 100 000 сокета)
- Висока производителност (до 1000 изпращания на задания в секунда и 600 изпълнение на задания в секунда)
- Силно конфигурируем с около 100 добавки
- Наличие на График за справедливо споделяне на ресурси
- Превантивно и групово планиране (отрязване на времето на паралелни задачи)
- Интегриране с база данни
- Разпределяне на ресурси, оптимизирано за мрежова топология и топология на възел (сокети, ядра и хипернишки)
- Предварителна резервация
- За всяко задание може да стартират различни операционни системи

- Поддръжа масиви от задачи
- Профилира работата (периодично взема проби от използването на процесора, използването на паметта, консумацията на енергия, използването на мрежата и файловата система на всяка задача)
- Поддържа MapReduce.

Наблюдаваното състояние на Суперкомпютъра включва: брой процесори, размер на оперативната памет, размер на временното дисково пространство и състояние (НАГОРЕ, НАДОЛУ и т.н.). Допълнителната информация за Суперкомпютъра включва тегло на задачата (предпочитание при разпределяне на работа) и функции (произволна информация като скорост или тип на процесора). Отделните сървъри в Суперкомпютъра са групирани в дялове. Информацията за дяла включва: име, списък на свързаните възли, състояние (НАГОРЕ или НАДОЛУ), ограничение за максимално време за работа, максимален брой възли за задание, списък за групов достъп, приоритет (важен, ако възлите са в множество дялове) и споделена политика за достъп до възел с опция ниво на свръхабонамент за групово планиране (напр. ДА, НЕ или НАСИЛА:2). Пример за конфигуриране на Slurm е показан по-долу:

```
#
# Sample /etc/slurm.conf
#
SlurmctldHost=linux0001  # Primary server
SlurmctldHost=linux0002  # Backup server
#
AuthType=auth/munge
Epilog=/usr/local/slurm/sbin/epilog
PluginDir=/usr/local/slurm/lib
Prolog=/usr/local/slurm/sbin/prolog
SlurmctldPort=7002
SlurmctldTimeout=120
SlurmdPort=7003
SlurmdSpoolDir=/var/tmp/slurmd.spool
SlurmdTimeout=120
StateSaveLocation=/usr/local/slurm/slurm.state
```

```
TmpFS=/tmp
#
# Node Configurations
#
NodeName=DEFAULT CPUs=4 TmpDisk=16384 State=IDLE
NodeName=lx[0001-0002] State=DRAINED
NodeName=lx[0003-8000] RealMemory=2048 Weight=2
NodeName=lx[8001-9999] RealMemory=4096 Weight=6 Feature=video
#
# Partition Configurations
#
PartitionName=DEFAULT MaxTime=30 MaxNodes=2
PartitionName=login Nodes=lx[0001-0002] State=DOWN
PartitionName=debug Nodes=lx[0003-0030] State=UP Default=YES
PartitionName=class Nodes=lx[0031-0040] AllowGroups=students
PartitionName=DEFAULT MaxTime=UNLIMITED MaxNodes=4096
PartitionName=batch Nodes=lx[0041-9999]
```

Литература

1. SLIURM Workload Manager, https://slurm.schedmd.com/documentation.html
2. What is Slurm and is it Still Relevant for Modern Workloads?, https://www.run.ai/guides/slurm#:~:text=Slurm%20is%20a%20system%20for,user%20to%20a%20compute%20node.
3. Deploy an HPC cluster with Slurm, https://cloud.google.com/hpc-toolkit/docs/quickstarts/slurm-cluster
4. How to test available network bandwidth using 'iperf', https://www.dell.com/support/kbdoc/en-bg/000139427/how-to-test-available-network-bandwidth-using-iperf
5. How to use iPerf3 to test network bandwidth, https://www.techtarget.com/searchnetworking/tip/How-to-use-iPerf-to-measure-throughput
6. Using iPerf to Test Network Speed and Bandwidth, https://woshub.com/testing-network-bandwidth-using-iperf/
7. HDFS NFS Gateway, https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsNfsGateway.html
8. Using the NFS Gateway for accessing HDFS, https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.5/data-storage/content/using_the_nfs_gateway_for_accessing_hdfs.html
9. Adding and Configuring an NFS Gateway, http://188.93.19.26/static/help/topics/admin_hdfs_nfsgateway.html