АСТРОФИЗИЧЕСКИЙ БЮЛЛЕТЕНЬ, 2015, том 70, № 2, с. 251–256

УДК 004.738.5; 004.771; 520.9

НАЧАЛЬНЫЙ ЭТАП ИССЛЕДОВАНИЯ СТЕНДА ПЕРЕДАЧИ БОЛЬШИХ ДАННЫХ ПО ПАРАЛЛЕЛЬНЫМ КАНАЛАМ. SDN ПОДХОД

© 2015 С. Э. Хоружников[1], В. А. Грудинин[1], О. Л. Садов[1], А. Е. Шевель[1,2], В. Б. Титов[1,3*], А. Б. Каирканов[1]

[1]Университет ИТМО, Санкт-Петербург, 197101 Россия

[2]Петербургский институт ядерной физики, Санкт-Петербург, 188300 Россия

[3]Санкт-Петербургский государственный университет, Санкт-Петербург, 199034 Россия

Поступила в редакцию 20 августа 2014 года; принята в печать 17 декабря 2014 года

Передача Больших Данных по компьютерным сетям — важная и неизбежная операция в прошлом, настоящем и обозримом будущем. Целый ряд разрабатываемых и уже работающих астрономических проектов связан с Большими Данными. Есть ряд методов передачи данных по глобальной сети (Интернет) с многообразными инструментами. В этой работе рассматривается передача фрагмента Больших Данных от одной точки Интернета к другой, в общем случае на далекое расстояние: порядка тысяч километров. Анализируются несколько свободно распространяемых систем передачи больших данных. Отмечаются важнейшие архитектурные свойства, и предлагается использовать методы протокола SDN OpenFlow для тонкой настройки передачи данных по параллельным каналам связи.

Ключевые слова: *инструменты виртуальной обсерватории*

1. ВВЕДЕНИЕ

Проблема Больших Данных[1] известна много лет. В разные периоды термин «Большие Данные» подразумевал разный объем и характеристики данных. Имея в виду «три V»: скорость (Velocity), объем (Volume), разнообразие (Variety), мы можем обратить внимание на то, что все эти черты относятся к текущему состоянию технологии. Например, в 80-х объем в 1 ТБ рассматривался как гигантский объем. В 1824 г. Чарльз Бэббидж получил золотую медаль Королевского астрономического общества «за свое изобретение машины для вычисления математических и астрономических таблиц» с беспрецедентной точностью, скорость этих вычислений соответствовала Большим Данным того времени. Разработка в 70–80-е гг. FITS-формата стандартизовала обмен изображениями между различными астрономическими учреждениями. В наши дни разрабатываются и используются системы, которые работают с Большими Данными в современном понимании. Стандарты доступа к большим массивам астрономических данных (стандарты для метаданных, форматов, языков запроса и т. д.), технологии работы, в том числе с Большими Данными, разрабатываются и поддерживаются международным альянсом IVOA (International Virtual Observatory Alliance), который был создан, чтобы «способствовать международной координации и сотрудничеству, необходимому для разработки и внедрения инструментов, систем и организационных структур, необходимых для обеспечения международного использования астрономических архивов как единой и функционирующей виртуальной обсерватории». Все проекты, работающие с Большими Данными, руководствуются рекомендациями IVOA. Таковы проекты ESO VLT, NOAO CTIO, NASA Kepler, NASA HMS и др.

Есть ряд сторон этой проблемы: хранение, анализ, передача и т. д. В этой статье мы обсуждаем один из важных аспектов Больших Данных: передачу по глобальной компьютерной сети.

2. ИСТОЧНИКИ БОЛЬШИХ ДАННЫХ

Известен длинный список видов человеческой деятельности (научной и деловой), которые являются генераторами значительного объема дан-

*E-mail: tit@astro.spbu.ru

[1]http://en.wikipedia.org/wiki/Big_data

ных [1–3], см. проекты SKA,[2] LSST,[3] FAIR,[4] ITER,[5] а также сайты CERN[6] и CLDS.[7]

Согласно [1], полный объем деловой переписки в мире в 2012 г. составил примерно 3000 ПБ (3×10^{18} Б). Общепринятая оценка полного объема сохраняемых данных с 2000 г. возрастает ежегодно в 1.5–2 раза. В этой работе (и для наших тестирований) мы будем полагать, что данные объемом около 100 ТБ (10^{14} Б) и более можно называть Большими Данными. Наверняка объем Больших Данных со временем будет расти.

Другая проблема Больших Данных — их сохранение в течение долгого времени: несколько десятилетий и более. Многие стороны нашей личной, общественной или деловой жизни и технические данные хранятся теперь в цифровой форме. Большие объемы таких данных нужно запоминать и хранить. Например, результаты медицинских тестов, данные, порожденные разного сорта значимыми двигателями (авиадвигатели, генераторы электростанций и т. д.), и другие данные должны архивироваться на долгое время. Хранимые данные будут содержаться в распределенных (локально и глобально) хранилищах. Принимается, что точные копии (replicas) хранимых данных должны храниться в нескольких местах на разных континентах, чтобы исключить их потерю в технических, природных или социальных бедствиях.

Исторически одной из первых областей, где появились Большие Данные, была физика высоких энергий. Был исследован ряд проблем передачи данных и решен целый круг задач. В настоящее время все больше научных и деловых областей имеют дело (или планируют это сделать) с Большими Данными [4]. Вот список разрабатываемых или уже работающих астрофизических/физических проектов [5–9]:

- Hipparcos, 1989–1992 гг., общий объем данных 300 ГБ;
- ESO VLT, с 1999 г., общий размер наблюдений 65 ТБ и растет на 15 ТБ в год;
- NASA Kepler,[8] с 2009 г., 100 ГБ в месяц;
- LOFAR (LOw Frequency ARray),[9] 2012 г., до 1 ПБ в сутки;
- Gaia (Global Astrometric Interferometer for Astrophysics), 1 ПБ в год;
- ПРАО (Пущино), все проекты, 10–100 ГБ в сутки;
- Радиоастрон, 1.28 ТБ в сутки;
- CERN, все проекты, 1 ПБ в сутки;
- LSST (Large Synoptic Survey Telescope), 2020 г., объем данных 10 ПБ в год;
- ITER (International Thermonuclear Experimental Reactor), 2020 г., 1–2 ПБ в сутки;
- CTA (Cherenkov Telescope Array), 2015–2020 гг., 20 ПБ в год;
- SKA (Square Kilometer Array), 2019–2024 гг., 1500 ПБ в год.

Четыре из приведенных выше проектов еще не окончены, и чем дальше срок окончания, тем больше данных планируется получить. В конце декабря 2013 г. вышел на планируемую работу Gaia, в настоящее время уже идет поток информации, который выйдет на уровень 1 ПБ в год.

Часто наблюдатели не имеют возможности долго хранить данные, и хранится только избранная их часть [10]. Для глубокого анализа требуется распределить полученные данные между участниками коллектива по всему миру. Это означает, что значительная часть экспериментальных данных должна передаваться по интернету.

3. СВОБОДНО ДОСТУПНЫЕ УТИЛИТЫ ПЕРЕДАЧИ ДАННЫХ ПО СЕТИ

Время передачи по глобальной компьютерной сети (Интернет) зависит от реальной пропускной способности канала связи и объема данных. Учитывая, что мы говорим об объемах в 100 ТБ и более, можно оценить минимальное требуемое время для копирования данных по сетевому каналу с пропускной способностью в 1 Гбит. Это дает нам около 100 МБ $с^{-1}$, следовательно, 100 ТБ/100 МБ $с^{-1}$ = $= 1\,000\,000$ с $= 277.^{h}8 = 11.^{d}6$. В течение этого времени параметры сетевого канала могут меняться. Например, может значительно варьироваться процент потерянных сетевых пакетов. Канал данных может страдать от прерываний операций на различное время: секунды, часы, дни.

Теперь давайте посмотрим на сетевые параметры ядра Linux. В директории `/proc` в Scientific Linux (клон RedHat) версии 6.5 имеется около полутысячи параметров, описывающих сетевой канал в ядре. Не все они одинаково чувствительны или оказывают влияние на процесс передачи данных. Наиболее важные из них — это размер окна TCP, MTU, алгоритм устранения перегрузки

[2] `http://skatelescope.org/`
[3] `http://www.lsst.org/lsst/`
[4] `http://www.fair-center.eu/`
[5] `http://www.iter.org/`
[6] `http://www.cern.ch/`
[7] `http://clds.sdsc.edu/`
[8] `http://www.nasa.gov/mission_pages/kepler/`
[9] `http://www.lofar.org/`

и т. д. Конечно, очень важно число независимых сетевых каналов, которые можно было бы использовать параллельно. Важны также такие сетевые параметры, как время прохождения сигнала туда и обратно (RTT) и процент потерянных сетевых пакетов. Наконец, понятно, что для достижения максимальной скорости передачи данных в каждой передаче данных большого объема нам нужно иметь возможность в течение процесса передачи настраивать (устанавливать) разное число потоков (threads), разный размер окна TCP и т. д.

Рассмотрим теперь свободно доступные утилиты передачи данных, которые можно использовать для передачи Больших Данных по сети.

3.1. Концепции сравнения утилит передачи данных

Приведём кратко характеристики для сравнения утилит передачи данных, которые могли бы помочь при передаче данных.

- Режим многопоточной передачи данных — способность использовать несколько потоков TCP параллельно.
- Режим многоканальной передачи данных — способность использовать более одного канала параллельно; это важное свойство, особенно если есть возможность учитывать, что доступные сетевые каналы не одинаковы по производительности и условиям (надежность, цена, реальное состояние и т. д.
- Возможность устанавливать параметры нижнего уровня, например, размер окна TCP и т. д.
- Метод обхода сетевых проблем (ошибок, задержек и т. д.). Другими словами, возможно ли продолжить передачу данных после перезапуска в случае сбоя при передаче?

По сути передача данных состоит из многих шагов: чтение данных из хранилища, передача данных по сети, запись полученных данных в хранилище на удаленной компьютерной системе. В этой работе наше внимание концентрируется в основном на сетевом процессе передачи.

3.2. Утилиты передачи данных низшего уровня

Отметим несколько утилит для передачи данных по сети (по крайней мере часть из них известны уже около десяти лет).

- Один из протоколов низшего уровня для передачи данных по сети — UDT.[10] UDT — это и библиотека, которая реализует протокол передачи данных, позволяющий использовать `udt`, а не `tcp`. В некоторых случаях библиотека может помочь улучшить использование канала данных, т. е. уменьшить время передачи.
- Протокол RDMA по конвергированному Ethernet (RDMA over Converged Ethernet, RoCE) изучался в [4]. Обнаружилось, что во многих случаях RoCE показывает лучшие результаты, чем UDP, UDT и стандартный TCP.
- MP TCP[11,12] — интересный протокол, который для одной передачи данных позволяет параллельно использовать несколько каналов данных. Протокол реализован как драйвер ядра Linux.
- Семейство (Open)SSH[13] — хорошо известные утилиты передачи данных, предоставляющие строгую аутентификацию и ряд алгоритмов шифрования данных. Возможно также сжатие данных до шифрования для уменьшения объема передаваемых данных. Есть две известные разновидности SSH: подкорректированная версия SSH,[14] которая может использовать увеличенный размер буферов, и SSH с аутентификацией GSI. Полноценного возобновления работы после сбоя нет. Нет и параллельных потоков передачи данных.
- bbcp[15] — утилита для смешанной передачи данных. Предполагается, что bbcp выполняется на обеих сторонах, т. е. передатчик — как клиент, и получатель — как сервер.
- Утилита bbftp[16] для передачи массивов данных. Она реализует свой собственный протокол передачи, который оптимизирован для больших файлов (больше чем 2 ГБ) и защищен, поскольку не читает пароль из файла и шифрует информацию о соединении.
- Xdd [11] — утилита, разработанная для оптимизации передачи данных и процессов ввода/вывода для систем хранения.
- fdp[17] — Java-утилита для многопоточной передачи данных.
- gridFTP[18] — усовершенствованная утилита передачи данных для инфраструктуры безопасности грид-систем (Grid Security Infrastructure, GSI).

Многие из них очень эффективны для передачи данных с точки зрения использования пропускной способности канала. Однако передача Боль-

[10] `http://udt.sourceforge.net/`

[11] `http://mptcp.info.ucl.ac.be/`

[12] `http://multipath-tcp.org/`

[13] `http://www.openssh.org/`

[14] `http://sourceforge.net/projects/hpnssh/`

[15] `http://www.slac.stanford.edu/~abh/bbcp/`

[16] `http://doc.in2p3.fr/bbftp/`

[17] `http://monalisa.cern.ch/FDT/`

[18] Там же.

ших Данных предполагает большое время передачи (несколько часов, дней или еще больше). Для длительных промежутков трудно положиться на такие простые процедуры передачи. Как мы уже упоминали, может измениться пропускная способность и процент потерянных пакетов в сетевом канале, может израсходоваться квота дискового пространства и так далее.

3.3. Службы передачи файлов среднего уровня

FTS3[19,20] — относительно новый и перспективный инструмент для передачи данных большого объема по сети. Имеет многие свойства из уже упомянутых и некоторые другие. Есть усовершенствованная возможность для отслеживания передачи данных (log), возможность использовать интерфейсы http, restful и CLI для контроля процесса передачи данных.

Другая интересная разработка — SHIFT,[21] которая предназначена для выполнения надежной передачи данных в LAN и WAN. Уделено много внимания надежности, усовершенствованию отслеживания, производительности передачи данных и использованию параллельной передачи данных между так называемыми эквивалентными хостами (между компьютерными кластерами).

3.4. Служба управления данными высокого уровня: PhEDEx

PhEDEx[22,23,24] (Physics Experiment Data Export) используется (и разрабатывается) в сотрудничестве по эксперименту Compact Muon Solenoid (CMS) [12, 13] в ЦЕРНе. Эксперимент дает большое количество экспериментальных данных (в 2013 г. было записано около 130 ПБ). Обработка данных требует их копирования на ряд больших компьютерных кластеров (около 10 мест в различных странах и на разных континентах) для анализа и архивирования. Позже части данных могут быть скопированы на меньшие вычислительные ресурсы (более, чем 60 мест). Полный объем передачи данных достигает 350 ТБ в сутки [13]. В ближайшем будущем ежедневный объем, возможно, будет расти. Поскольку между несколькими сайтами может быть более одного канала, в PhEDEx разработан метод маршрутизации, который позволяет проверить другой маршрут, если умалчиваемый маршрут недоступен.

Наконец, система PhEDEx является очень сложной, и служба управления зависит от среды кооперации физических экспериментов. Маловероятно, что PhEDEx можно использовать в другой среде без модификации.

4. ОБСУЖДЕНИЕ

Отмеченные утилиты имеют несколько полезных для передачи данных свойств:

- все утилиты имеют архитектуру клиент–сервер;
- способны устанавливать размер буфера, размер окна TCP и т. д.;
- имеют возможность выполнять различные операции до реальной передачи данных и после передачи данных, например, сжатие/восстановление, использовать ряд драйверов/методов для чтения/записи файлов на вспомогательную память;
- могут использовать ряд методов аутентификации;
- могут использовать для передачи данных более одного потока, более одного сетевого канала;
- могут использовать несколько алгоритмов аутентификации;
- могут использовать ряд методов, чтобы сделать передачу данных надежнее;
- утилиты не одинаковы по числу параметров и сфере решаемых задач, часть из них хорошо подходит для использования в качестве независимых утилит передачи данных почти во всех средах, другие, как PhEDEx (в CMS) и аналогичные системы в кооперации ATLAS,[25] предназначены для использования как часть более сложной и специфической компьютерной среды.

Другими словами, есть набор инструментов, которые во многих случаях могут помочь передать Большие Данные по сети. Ряд утилит может использовать более одного сетевого канала, увеличивая тем самым скорость передачи данных.

Вместе с тем не предлагается никаких средств тонкой настройки параллельных каналов данных. Тонкая настройка рассматривается как возможность применить различные правила к различным каналам данных. Вообще говоря, параллельные каналы данных могли бы быть совершенно различными по природе, свойствам и условиям использования. В частности, при передаче данных

[19] http://www.eu-emi.eu/products/-/
/asset_publisher/1gkD/content/fts3
[20] https://svnweb.cern.ch/trac/fts3
[21] http://fasterdata.es.net/data-transfer-tools/
[22] https://cmsweb.cern.ch/phedex
[23] https://twiki.cern.ch/twiki/bin/view/CMSPublic/
/PhedexAdminDocsInstallation
[24] http://hep-t3.physics.umd.edu/HowToForAdmins/
/phedex.html
[25] http://rucio.cern.ch/

предполагается использование QoS для каждого сетевого канала и способность оперативно, в процессе работы, изменять правила. Все это приводит к мысли, что необходимо специальное приложение, которое могло бы следить за состоянием каналов данных и изменять параметры передачи согласно реальной ситуации в каналах данных. Параметры сетевого канала планируется устанавливать, используя протокол OpenFlow.[26] [14] Чтобы следить за состоянием каналов данных, будет использовано специальное средство PerfSonar [15].

Очевидно, что требуется специализированная установка для теста, чтобы исследовать процесс передачи данных с помощью описанных утилит и оборудования. Настраиваемый испытательный стенд должен иметь возможность моделировать по крайней мере основные сетевые проблемы, например, изменение RTT, задержки, процент утраченных пакетов и т. д. Разработка такого стенда начата в университете ИТМО в Лаборатории сетевых технологий в распределенных вычислительных системах.[27] Это направление работы привлекает многих исследователей [16].

Ожидается, что стенд будет платформой для сравнения различных утилит в одной и той же среде. Как первый шаг, планируется выполнить сравнительные измерения с набором утилит передачи данных, подробно записывая все условия измерений. Это позволит в будущем сравнить на стенде другие методы передачи данных точно в такой же среде.

5. РАБОТА СТЕНДА

Стенд состоит из двух серверов HP DL380p Gen8 E5-2609, Intel(R) Xeon(R) CPU E5-2640 @2.50GHz, 64 GB под управлением Scientific Linux 6.5. Поскольку планируется все тестировать в виртуальной среде, для каждой упомянутой системы передачи данных используется две виртуальные машины (VM): одна VM — как передатчик, а другая — как приемник. Другими словами, у нас есть около десяти VM. Чтобы соединить эти VM, развернута и запущена платформа OpenStack.[28] Также развернута PerfSonar.

Для изучения различных типов данных была разработана специальная процедура, генерирующая тестовую директорию с файлами произвольной длины, полный объем тестовой директории определяется параметром процедуры. При генерации тестовых данных можно задать среднее значение и дисперсию размера файла. Данные в каждом файле тестовой директории намеренно готовятся так, чтобы исключить возможный эффект сжатия данных (если таковое есть) при передаче данных.

На начальной стадии планируется сравнить все приведенные системы передачи данных в локальной сети, чтобы убедиться, что всё (все скрипты) функционирует должным образом. Отдельная проблема — записать во время измерений все журналы, параметры и т. д.. В частности, это подразумевает требование автоматически записывать всю директорию `/proc` в некоторое место, скажем, «директорию log». Также требуется записать все параметры и сообщения движка/утилиты передачи данных. Наконец, состояние канала данных также предполагается записывать. Вся упомянутая информация должна сохраняться в «директории log». Очевидно, все должно осуществляться скриптами, предназначенными для выполнения измерений.

Разработанные процедуры (скрипты) и краткие описания доступны в Интернете.[29]

6. ЗАКЛЮЧЕНИЕ

При планировании проекта, который связан с большим объемом экспериментальных данных, важно учесть затраты на передачу данных по сети. Можно указать несколько пунктов в наблюдательном цикле, где передача Больших Данных по сети является реальной необходимостью:

- сбор данных;
- быстрый контроль данных (и/или фильтрация);
  - возможная передача данных (может быть локальной или удаленной);
- сохранение данных во внешней памяти;
  - возможная передача данных на удаленный вычислительный центр (может быть в несколько пунктов) для дальнейшего анализа;
- анализ данных.

В этой работе описывается только техника передачи данных, которая является неотъемлемой частью наблюдательного цикла. Ясно, что в предстоящих экспериментах, где ожидается громадный поток данных, чем эффективнее передача данных, тем продуктивнее научные исследования.

БЛАГОДАРНОСТИ

Работа поддержана Санкт-Петербургским национальным исследовательским университетом информационных технологий, механики и оптики.

[26] `https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf`

[27] `http://sdn.ifmo.ru/`

[28] `http://www.openstack.org`

[29] `https://github.com/itmo-infocom/BigData`

СПИСОК ЛИТЕРАТУРЫ

1. J. Pearlstein, http://www.wired.com/magazine/ /2013/04/bigdata/
2. L. Borovick and R. L. Villars, http:// /unleashingit.com/docs/B13/Cisco%20UCS/ /critical_big_data_applications.pdf
3. W. E. Johnston, E. Dart, M. Ernst, and B. Tierney https://tnc2013.terena.org/getfile/402/; https://tnc2013.terena.org/getfile/716/
4. B. Tierney, E. Kissel, M. Swany, and E. Pouyoul http://www.es.net/assets/pubs_presos/ /eScience-networks.pdf
5. M. Juric, J. Kantor, T. S. Axelrod, et al., American Astron. Soc. Meeting Abstracts, No. 221, 247.01 (2013).
6. P. Dewdney, W. Turner, R. Millenaar, et al., SKA1 System Baseline Design, SKA-TEL-SKO-DD-001.
7. B. S. Acharya, M. Actis, T. Aghajani, et al., Astroparticle Phys. **43**, 3 (2013).
8. P. de Teodoro, A. Hutton, B. Frezouls, et al., in *Astrostatistics and Data Mining*, Ed. by L. M. Sarro, L. Eyer, W. O'Mullane, and J. De Ridder, Springer Ser. in Astrostatistics **2**, 107 (2012).
9. Е. А. Исаев, В. В. Корнилов, П. А. Тарасов и др., Препринт № 8 (Физический институт им. Лебедева РАН, Москва, 2014).
10. S. Karpov, G. Beskin, S. Bondar, et al., Acta Polytechnica **53**, 38 (2013).
11. S. W. Hodson, S. W. Poole, T. M. Ruwart, and B. W. Settlemyer, http://info.ornl.gov/sites/ /publications/files/Pub28508.pdf
12. The CMS Collaboration, J. Instrumentation **3**, S08004 (2008).
13. R. Kaselis, S. Piperov, N. Magini, et al., J. Phys. Conf. Ser. **396**, 042033 (2012).
14. B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, et al., Communications Surveys and Tutorials, IEEE **16**, 1617
15. J. Zurawski, S. Balasubramanian, A. Brown, et al., http://www.es.net/assets/pubs_presos/ /20130910-IEEE-BigData-perfSONAR2.pdf
16. D. Gunter, R. Kettimuthu, E. Kissel, M. Swany, et al., in *Proc. Meeting on High Performance Computing, Networking Storage, and Analysis (2012 SC Companion), Salt Lake City, USA, 2012*, p. 1600.

Initial-Stage Examination of a Testbed for the Big Data Transfer over Parallel Links. The SDN Approach

S. E. Khoruzhnikov, V. A. Grudinin, O. L. Sadov, A. E. Shevel, V. B. Titov, and A. B. Kairkanov

The transfer of Big Data over a computer network is an important and unavoidable operation in the past, present, and in any feasible future. A large variety of astronomical projects produces the Big Data. There are a number of methods to transfer the data over a global computer network (Internet) with a range of tools. In this paper we consider the transfer of one piece of Big Data from one point in the Internet to another, in general over a long-range distance: many thousand kilometers. Several free of charge systems to transfer the Big Data are analyzed here. The most important architecture features are emphasized, and the idea is discussed to add the SDN OpenFlow protocol technique for fine-grain tuning of the data transfer process over several parallel data links.

Keywords: *virtual observatory tools*