

# Астрономическое программное обеспечение Unix

## Небесная механика

В. Титов  
[tit@astro.spbu.ru](mailto:tit@astro.spbu.ru)

Научно-исследовательский Астрономический институт  
им. В.В.Соболева

Программирование, 3 курс, [21.11.2017](#)

# Почему Unix

Открытая система . . . разработанная  
в соответствии с общедоступными и  
общепринятыми стандартами.

- переносимость;
- интероперабельность;
- масштабируемость;
- доступность

Научный результат должен быть

- воспроизводимым
- доступным

# Почему Python

## Традиционные Астрономические приложения

Закрытый/не используемый совместно

Разобщенный & Дублирующий

Нисходящее планирование

Проектирование, ориентированное на комитет

Бесконечный анализ & спор

Нежелающее ничего обсуждать старые техники

Нет руководителя для разрешения конфликтов

## Python и Open Source

Контактный/используемый совместно

Используемый в общих проектах

Восходящее/Свободная структура

Проектирование «исполнителями»

Настроенный на действие & эксперимент

Хорош для замены старых техников

BDFL разрешает конфликты

# Unix Way, или Философия Unix

Doug McIlroy:

- Пишите программы, которые делают что-то и делают это хорошо.
- Пишите программы, которые работали бы вместе.
- Пишите программы, которые поддерживали бы текстовые потоки, поскольку это универсальный интерфейс.

# Unix Way, или Философия Unix

Rob Pike (стиль программирования на C):

- вило 1 Вы не знаете, как распределяется время выполнения программы. Программа может работать медленнее в самых неожиданных местах, поэтому не стройте догадки и не старайтесь сделать программу быстрее до тех пор, пока не докажете, что узкое место именно здесь.
- вило 2 Измеряйте. Не начинайте оптимизировать скорость до тех пор, пока ваш профилировщик не укажет вам, какие участки кода являются наиболее медленными.  
(«Оптимизация — корень всех зол». Д.Кнут)
- вило 3 Изощрённые алгоритмы работают медленнее при малых объёмах входных данных. Значение константы  $C$ , используемой для определения асимптотического поведения алгоритма, у них обычно велико, а объёмы входных данных обычно малы. Поэтому прибегайте к изощрённым алгоритмам только если с помощью Правила 2 вы обнаружите, что объёмы входных данных для них оказываются достаточно велики.

# Unix Way, или Философия Unix

Rob Pike (стиль программирования на C):

- правило 4 Упрощайте ваши алгоритмы и структуры данных где это возможно, поскольку изощрённые алгоритмы труднее реализовать без ошибок. Структуры данных для большинства программ могут быть основаны на массивах, связных списках, хеш-таблицах и двоичных деревьях. (Правила 3 и 4 Кен Томпсон сформулировал так: «Если сомневаетесь, ищите правильное решение путем перебора всех возможных». В инженерной философии это KISS: Keep It Simple, Stupid.)
- правило 5 Данные господствуют. При правильной и хорошо организованной структуре данных, алгоритмы становятся очевидными. Структуры данных, а не алгоритмы, являются центральной частью в программировании.  
(«Пиши тупой код, который использует умные данные»).
- правило 6 Правила 6 не существует.

# Астрономия. Линукс

В дистрибутивах (репозиториях) Linux много астрономических программ. В дистрибутиве Fedora 20 около сотни.

Все астрономические программы можно разделить на

- программы общего назначения;
- профессиональные;
- исследовательские;
- компоненты виртуальной обсерватории.

Разделение довольно условно

# Программы общего назначения

- планетарии;
- обучающие программы;
- моделирующие программы;
- библиотеки.

# Планетарии

- XEphem — первый планетарий в Linux;
- KStar — планетарий для KDE;
- **Stellarium** — планетарий в режиме реального времени.

# Планетарии. Каталоги, доступные в XEphem

Во список каталогов, доступных в XEphem:

- General Purpose:
  - SKY2000 Master Catalog v.4, 2002
  - SKY2000 Bright Stars Subset, 2002
  - ...
- Field Stars:
  - GSC 2.2.0.1 Catalog, M18.5
  - GSC-ACT Catalog, M15
  - Hipparcos and Tycho-2 Catalogues
  - ...
- Galaxies and Clusters of Galaxies:
  - HYPERLEDA Catalogue, 2003 (p.1, bright)
  - HYPERLEDA Catalogue, 2003 (p.2, dim)
  - ...

Всего 476470660 объектов. (15 лет.)

- **Nightfall** — анимационное изображение затменных двойных;
- **NOVA** — интегрированная наблюдательная среда;
- **Kali** — численное интегрирование задачи  $N$  тел;
- **Skymap** — полезные астрономические отображения.

# Моделирование

- [Celestia](#) — виртуальное моделирование Вселенной в режиме реального времени (STA);
- [OpenUniverse](#) — моделирование тел солнечной системы в 3D;
- [wmMoonClock](#), [XVMoontool](#), [Xtide](#) — лунные эфемериды и другая информация о Луне, приливах...

# Библиотеки

- GSL, CFITSIO, LAPACK, Atlas, FFTW.
- SLALIB — часть проекта Starlink, библиотека для астрометрических вычислений;
- Библиотека исходных кодов астрофизики — коллекция ссылок на численные физические модели;
- Исходные коды для астрономического и численного программного обеспечения — коллекция С-кодов;
- Вычисление положения планет;
- CCD астрономия и Linux.

- Обработка наблюдений;
- Библиотеки;
- Моделирование;
- Справочные;
- Решение отдельных задач.
- Обработка и анализ изображений: MIDAS, IRAF, AIPS;
- Работа с FITS-форматами; ds9, fv
  - просмотр FITS изображений;
  - редактирование FITS файлов.
- Задачи глобального позиционирования (GPS).

# Библиотеки и программы

- Starlink/SLALIB (позиционная астрономия);
- Собрание исходных кодов для решения астрофизических задач, исходные тексты нескольких десятков реальных астрофизических задач;
- Исходные коды астрономических и численных программ для решения астрофизических задач;
- Вычисление положений планет;
- Работа с CCD матрицами;
- pyephem;
- ATpy;
- NumPy, Scipy, PyRAF, PyFITS.

# Моделирование

- **NBODY6** — интегрирование задачи  $N$  тел, Aarseth;
- **NEMO** — моделирование задач звездной динамики, Barnes, Hut, Teuben;
- **STARLAB** — после NEMO, Hut, McMillan, Makino и другие наследники NEMO;
- **Art of Computational Science** — Hut & Makino, электронный учебник;
- **GADGET** — моделирование на основе tree-кода с учетом гидродинамики, V.Springer, N.Yoshida;
- **PMFAST** — задача  $N$  тел (частица-решетка);
- **ORSA** (Orbit reconstruction, simulation and analysis) — конструирование, моделирование и анализ орбит;
- **Swift** — симплектический интегратор, изучение эволюции солнечной системы.

# Исследовательские задачи

- Получение результата не гарантировано;
- Алгоритмы только обкатываются;
- Нет и не может быть программ, делающих то, что нужно;
- Проверить то или иное предположение нужно как можно скорее...

Восьмерка...

# Ресурсы

- Linux for Astronomy CDROM >6 Гб;
- Проект ESO: [Scisoft](#) (IRAF, MIDAS, Eclipse, IDL, Gildas);
- Проекты [sourceforge.net](#);
- GPSTk [www.gpstk.org/sourceforge.org](http://www.gpstk.org/sourceforge.org);
- Github <https://github.com>;
- ...

- Web

- ИПА РАН (<http://www.ipa.nw.ru>)
- Пулково (<http://www.gao.spb.ru/personal/neo/>)
- ГАИШ (<http://www.sai.msu.ru/neb/>)
- IAU Minor Planet Center (<http://www.minorplanetcenter.org/>)
- ESO (<http://www.eso.org/>)
- NASA (<http://www.nasa.gov/>)
- NASA, JPL (<http://jpl.nasa.gov/>)
- NASA, JPL, SSD (<http://ssd.jpl.nasa.gov/>)

- GUIDE
- ЭРА
- ЭПОС
- CAS: Maxima (Macsyma)
- Gnuplot, pgplot и т. д.
- PSP
- **TeX**
- CVS, Git (см. например,  
<http://gpstk.svn.sourceforge.net/viewvc/gpstk/>)

# Виртуальные обсерватории

## ВСЕ АСТРОНОМИЧЕСКИЕ АРХИВЫ НА ВАШЕМ КОМПЬЮТЕРЕ

Цель ВО:

- Все астрономические данные должны работать, как единое целое
- Все центры данных предоставляют доступ к данным и инструменты анализа, визуализации, размещения и передачи данных
- Стандартизация данных и метаданных,
- Стандартизация методов обмена данными,
- Использование реестра (перечислены службы, что можно сделать с их помощью).

# Виртуальные обсерватории

- **Поиск изображений:** [Aladin](#), Datascope, SkyView, VODesktop, Data Discovery Tool, AladinLite
- **Поиск Спектров:** [Aladin](#), CASSIS, Datascope, [SPLAT](#), [Specview](#), VOServices, [VOSpec](#), Data Discovery Tool
- **Поиск Каталогов:** [Aladin](#), Datascope, [TOPCAT](#), [VirGO](#), VODesktop, Data Discovery Tool
- **Поиск Временных Рядов:** Time Series Search Tool
- **Визуализация изображений:** [Aladin](#), SkyView
- **Визуализация Спектров:** CASSIS, [SPLAT](#), Specview, VOServices, [VOSpec](#)
- **Визуализация Каталогов:** [Aladin](#), [TOPCAT](#), VOPlot, [VirGO](#)
- **Кросс-корреляция:** [Aladin](#), STILTs, [TOPCAT](#), CDS Xmatch Service, Cross Comparison Tool
- **Scatter, 3D-графика и гистограммы:** [TOPCAT](#), [VOPlot](#)
- **Статистика:** VOStat
- **Преобразование формата таблиц:** [TOPCAT](#), [VOConvert](#), [STILTS](#)
- **Запрос Баз Данных:** Selest, [TOPCAT](#), TAPHandle, TAPsh
- **Служба Footprint:** [Aladin](#), VOServices
- **Фильтрация кривых:** VOServices, Filter Profile Service
- **SED building:** VOSA, VOSED, [VOSpec](#), Iris
- **Fixing WCS:** [Aladin](#), WCSFixer

# Пример

Анализ отображений: [Aladin](#), Поиск каталогов: [Topcat](#)



# Как это устроено

ВО:

- Множество сервисов или служб данных. Все они подчиняются одним стандартизованным правилам. Инструменты (Aladin, Topcat), подчиняясь правилам могут получить данные от служб.
- Каждая служба зарегистрирована в одном из нескольких реестров. Элементы реестра это описательные метаданные. Что за служба, как до него добраться, каковы параметры запроса, каков результат.
- Службы возвращают данные в стандартных форматах (JPEG, FITS, VOTable).
- Все инструменты могут взаимодействовать друг с другом (SAMP).
- Службы могут быть различными, например, cone-search, службы доступа к таблицам, изображениям, спектрам, поиск реестров...
- Регистрация.
- Службы удаленного запуска.

# ВО с точки зрения разработчика

- ВО базируется на двух ключевых понятиях: Ресурсы и Службы.
- Документация IVOA.
- Набор протоколов IVOA определяет основные методы доступа для различных типов ресурсов данных.
- Данные обычно возвращаются в формате VOTable.
- В реестре каждый ресурс имеет идентификатор, метаданные перечисляют элементы, необходимые для описания ресурса.

# Требования к подготовке табличных данных

- Данные описываются достаточно точно для недвусмысленной интерпретации данных и понимания контекста, в котором получаются и обрабатываются данные, для этого предназначен ascii-файл ReadMe;
- данные представляются в формате, который позволяет использовать их текущими средствами, а именно *простыми текстовыми файлами*.

Почему не FITS:

- табличные данные легче сравнить с опубликованными данными;
- описание структуры таблицы в FITS хорошо приспособлено к использованию в компьютере, но чтение этого описания не очень приятно для чтения человеческим глазом;
- из-за фиксированной длины записи FITS-файлы, вообще говоря, больше, чем стандартные текстовые файлы, даже для бинарных FITS;
- из-за этой составной структуры файлов нельзя применять к FITS-файлам такие средства UNIX, как grep, sort, join, paste, awk.

# VOTable

В [VOTable](#) метаданные и данные хранятся отдельно.

Таблицы могут локальными или удаленными (это XML), сжатыми или нет.

XML позволяет сделать данные не только переносимыми, но и распределенными в сети Интернет.

# VOTable

Вот пример VOTable-документа:

```
<?xml version="1.0"?>
<!DOCTYPE VOTABLE SYSTEM "http://us-vo.org/xml/VOTable.dtd">
<VOTABLE version="1.0">
  <DEFINITION>
    <COOSYS ID="myJ2000" equinox="2000." epoch="2000." system="eq_FK5"/>
  </DEFINITION>
  <RESOURCE>
    <PARAM name="Observer" datatype="char" arraysize="*" value="William Herschel">
      <DESCRIPTION>This parameter is designed to store the observer's name
      </DESCRIPTION>
    </PARAM>
  <TABLE name="Stars">
    <DESCRIPTION>Some bright stars</DESCRIPTION>
    <FIELD name="Star-Name" ucd="ID_MAIN" datatype="char" arraysize="10"/>
    <FIELD name="RA" ucd="POS_EQ_RA" ref="myJ2000" unit="deg"
      datatype="float" precision="F3" width="7"/>
    <FIELD name="Dec" ucd="POS_EQ_DEC" ref="myJ2000" unit="deg"
      datatype="float" precision="F3" width="7"/>
    <FIELD name="Counts" ucd="NUMBER" datatype="int" arraysize="2x3x*"/>
  <DATA>
    <TABLEDATA>
      <TR>
        <TD>Procyon</TD><TD>114.827</TD><TD> 5.227</TD>
        <TD>4 5 3 2 1 2 3 4 6</TD>
      </TR>
      <TR>
        <TD>Vega</TD><TD>279.234</TD>
        <TD>38.782</TD><TD>8 7 8 6 8 6</TD>
      </TR>
    </TABLEDATA>
  </DATA>
```

# VOTable

Данные в VOTable представляются в одном из трех форматов: TABLEDATA, FITS и BINARY. Что такое TABLEDATA ясно из примера. FITS — формат, широко используемый и разработанный астрономами для передачи данных еще в семидесятых годах. FITS-файл можно или прямо включить в <FITS>...</FITS>, или предварительно восстановить метаданные. В формате BINARY содержатся просто двоичные данные (поток битов).

# VOTable DTD

А вот фрагмент DTD для VOTable:

```
<!-- DOCUMENT TYPE DEFINITION for VOTable =
      Virtual Observatory Tabular Format -->

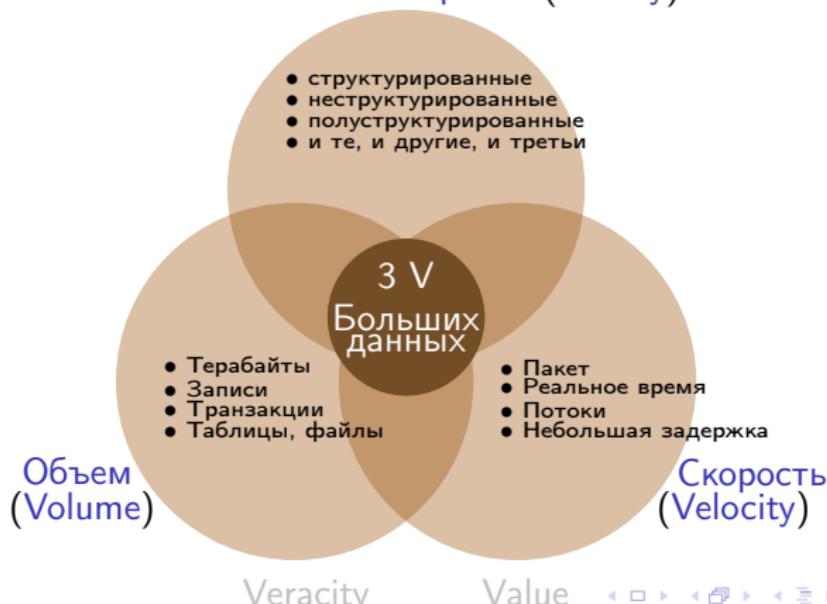
<!-- VOTABLE is the root element -->
<!ELEMENT VOTABLE (DESCRIPTION?, DEFINITION?, INFO*, RESOURCE*)>
<!ATTLIST VOTABLE
  . . . . .
>
<!-- RESOURCE can contain other RESOURCE -->
<!ELEMENT RESOURCE (DESCRIPTION?, INFO*, COOSYS*, PARAM*, LINK*,
  TABLE*, RESOURCE*)>
  . . . .
<!ELEMENT DESCRIPTION (#PCDATA)>
```

- Размер файлов > X ХБ (МБ, ГБ, ТБ, Пб, Эб, 36, Йб,  $10^{88}$ , гугол)?
- Такие данные, с которыми мы не можем справиться, но хотели бы...

# Что такое **БОЛЬШИЕ ДАННЫЕ?**

- Определяются быстрым ускорением увеличивающегося объема высокоскоростных, сложных и разнообразных данных. Большие Данные часто определяются по трем измерениям: **объем**, **скорость** и **разнообразие**.

## Разнообразие (Variety)



# Что такое **БОЛЬШИЕ ДАННЫЕ?**

- Любое количество информации, которое слишком велико, чтобы обрабатываться одним компьютером.
- Самое простое определение «Больших Данных» такое: «они не втискиваются в „Excel“».
- Каждый день создается 2.5 квинтильона ( $10^{18}$ ) байт данных, Более 90% данных в мире получено только в последние два года.
- Данные Большие, если объем данных это часть проблемы.
- Современная версия Большого Брата. Поиски онлайн, покупки со склада, посты Фейсбука, твиты или регистрации, использование сотового телефона, и т. д. создают потоки данных, которые, будучи организованы, классифицированы и проанализированы, раскрывают тренды и привычки нас самих и общества в целом.
- Трудно работать с данными с помощью имеющихся средств. Проблемы включают в себя сбор данных, сопровождение, хранение, поиск, коллективное использование, передачу, анализ и визуализацию.
- В настоящее время **Большие Данные** это синоним таких технологий как **Hadoop** и класса баз данных «**NoSQL**».

# Источники Больших Данных

- Hipparcos: 1989—1992 гг., общий объем данных 300 ГБ;
- ESO/VLT: с 1999 г., общий размер 65 ТБ + 15 ТБ/год;
- NASA/KEPLER: с 2009 г., 100 ГБ/мес;
- LOFAR: LOw Frequency ARray, 2012 г., до 1 ПБ/день;
- GAIA: Global Astrometric Interferometer for Astrophysics, 2014, <http://sci.esa.int/gaia/>, 200 ТБ/год—1 ПБ/год
- ПРАО (Пущино): все проекты, 10-100 ГБ/день;
- Радиоастрон: 1.28 ТБ/день;
- ЦЕРН: <http://www.cern.ch>, 20 ПБ/год (обрабатывается 1 ПБ/день);
- Широкоугольный обзорный телескоп-рефлектор (Large Synoptic Survey Telescope, LSST, ≈ 2020) <http://www.lsst.org>, 15 ТБ/ночь (10 ПБ/год)
- Международный экспериментальный термоядерный реактор (International Thermonuclear Experimental Reactor, ITER, ≈ 2020) <http://www.iter.org>, 1 ПБ/год;
- Решетка черенковских телескопов Cherenkov Telescope Array, CTA, ≈ 2015–2020 <http://www.cta-observatory.org/>, 20 ПБ/год
- Радиоинтерферометр «Квадратная километровая решетка» (Square Kilometre Array, SKA, ≈ 2019–2024) <https://www.skatelescope.org/>, 300-1500 ПБ/год

# Рост сетевого трафика

## Тенденция роста трафика ESnet

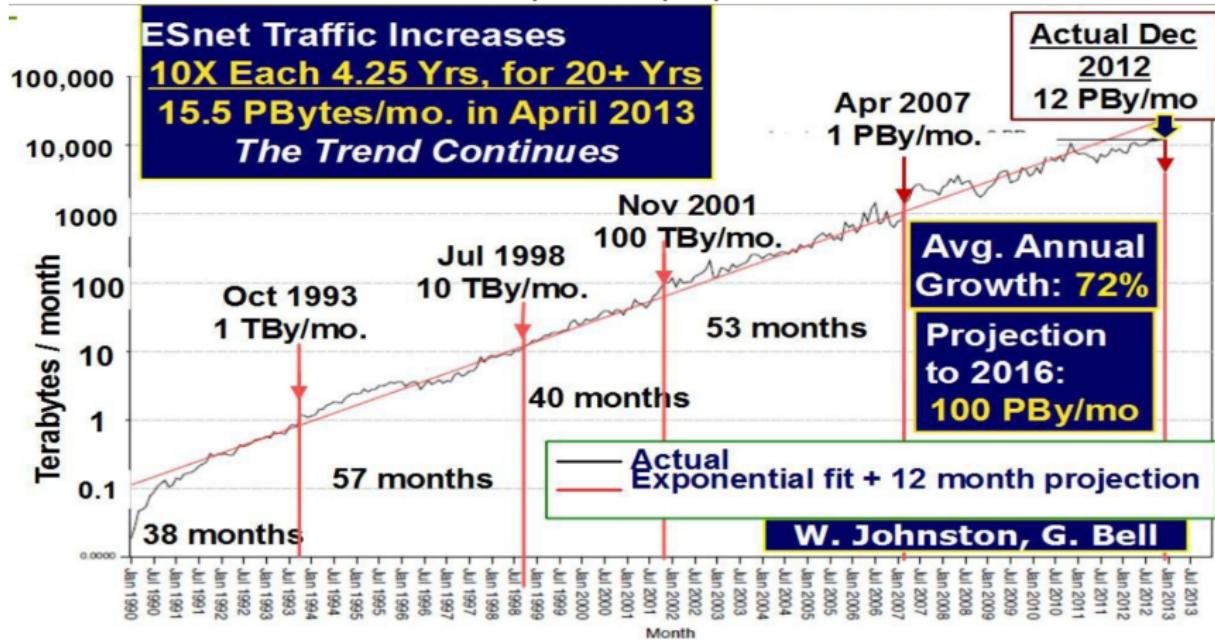


График ежемесячного входного трафика ESnet,  
январь 1990 – декабрь 2012

# Особенности передачи Больших Данных

- Передача может занять много часов или дней (при пропускной способности в 1 Гбит:  $100 \text{ ТБ}/100 \text{ МБ} = 1000000$  секунд или **11.6 дней**).
- Обстановка в каналах может измениться: время прохождения сигнала (RTT), % потерянных сетевых пакетов, пропускная способность канала данных.
- Наконец, может случиться сбой в работе канала данных (часы?, дни?).

## Grid-системы

100 ТБ/100 МБ=1 000 000 секунд или **11.6 дней**

Лучше [приложение скопировать к данным?](#)

Grid-система. Концепция разработана еще в 1999 г.

Создан набор служб, обеспечивающих надежный доступ к распределенным вычислительным ресурсам: компьютерам, кластерам, ресурсным центрам, центрам обработки данных, сетям, научной аппаратуре и т. д.:

- сайты (Вычислительный элемент, CE, Элемента хранения SE),
- серверы (Рабочий узел, WN, Пользовательский интерфейс, UI, Брокер ресурсов),
- службы (Поиск реплик файлов RLS, Информационный сервис IS, Подбор нужных CE и SE, Match-Maker-Broker)

Задание на управление — JDL.

# Grid-данные

Проект, касающийся данных должен обеспечивать

- создание распределенной файловой системы ( $>N$  петабайт),
- параллельный ввод/вывод и параллельную обработку,
- глобальную аутентификацию и контроль за доступом,
- глобальное управление ресурсами и для всех узлов кластера,
- разделение данных между отдельными центрами и эффективный доступ,
- совместное использование,
- мониторинг и администрирование,
- устойчивость к сбоям.

## Другие системы

Последовательный файл —→

(реляционные) базы данных —→

...

При увеличении объема данных вычисления должны перемещаться ближе к данным.

NOSQL, ключ–значение

MapReduce

# MapReduce

- Большие данные при обработке делятся на параллельные куски,
- Каждый кусок обрабатывается независимо,
- Результаты собираются в одно целое.

Преимущества в задачах:

- распределенной сортировки,
- поиска по образцу,
- индексации,
- машинного обучения.

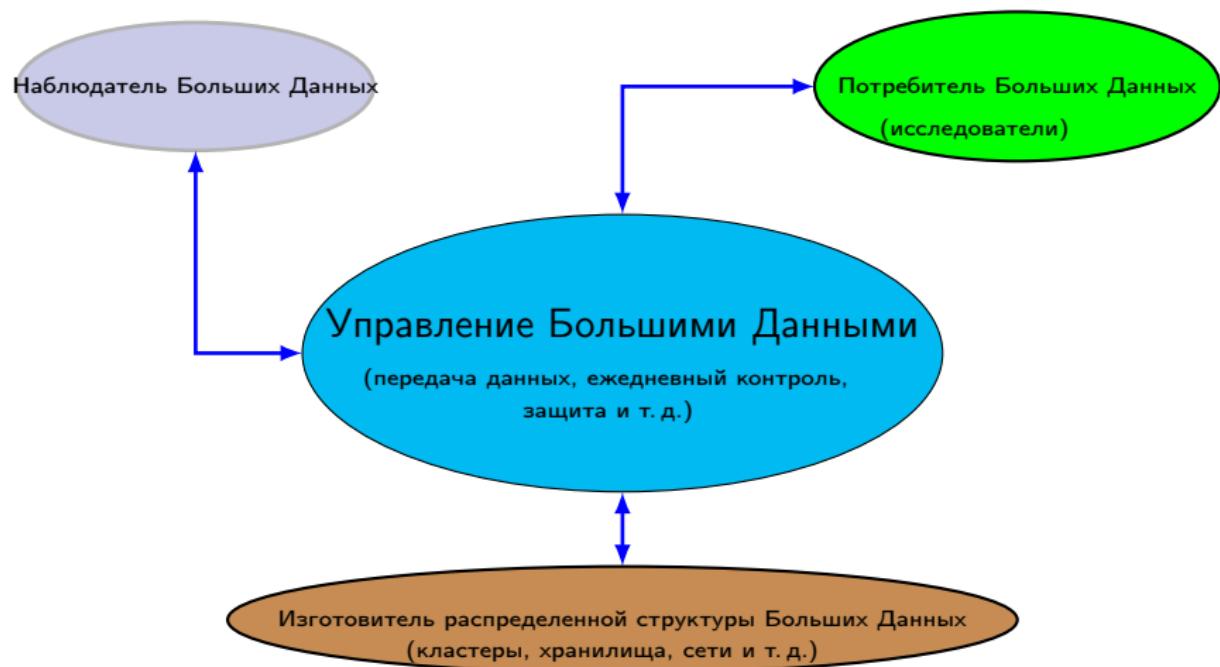
Недостатки: глобальное состояние, маленькие данные.

# Облака



# Большие Данные

- Хранение
- Доступ
- Обработка



# Информационные системы в фундаментальной науке БОЛЬШИЕ ДАННЫЕ



- BigData — создание, хранение и доступ
- программно-аппаратные средства для BigData
- динамически изменяющиеся BigData
- BigData в cloud-инфраструктуре
- когнитивные технологии анализа и мониторинга BigData

# N-Body

Конец

