Здравейте колеги,

Във връзка с проведеното упражнение на 4 Април, ето обобщение на заниманията ни, както и инструкции за стъпките, които изпълнихме:

0. Както може би се убедихме всички, вариантът на работа директно върху Windows 2003 чрез Cygwin, удари на камък. Простете за изгубеното време (което може би, все пак, е полезен опит).

1. Снабдете се с VMware image-а на Fedora Core 6, 32-битова инсталация, или от въпросната апокрифна flash-ка, която разпространяваме, или от share-а в зала 314: //St-314-d11/share. Самият image е zip файл с име fedora-fc6-i386. Има го и тук:

http://dl.dropbox.com/u/25250573/fedora-fc6-i386.zip Разархивирайте го. По default image-ът работи с около 1.2 GB RAM, което както видяхме, не е проблем за машините в компютърните зали на ФМИ.

2. Отворете този image с VMware player/Workstation и го заредете. Трябва да се стартира Fedora сесия с графичен интерфейс Gnome. Log-нете се като user:user и password:mathematics. За всеки случай, администраторският account е root/thoughtpolice.

3. На Desktop-а Ви ще откриете папка с име Installation, в която има разархивиран Hadoop 0.20.2. За IDE, използвайте eclipseEuropa папката! Има и папка eclipse, която е с версия Helios на платформата, но с нея ще имате проблеми при използването на Hadoop plug-in-а.

4. Тук е момента да се извиним на онези от Вас, които не са много comfortable с Java езика. Hadoop и HDFS са писани на Java, и затова MapReduce примерите, с които ще се занимаваме, са изцяло в термините на Java, както е и клиентското API на Hadoop. Моля питайте, ако нещо не е ясно, и на някоя стъпка се затруднявате.

5. На image-а ма форматиран HDFS клъстер, което става чрез следната shell команда:

$> bin/hadoop namenode -format

, изпълнена в hadoop-0.20.2 папката (тя вече е изпълнена)

6. На тази стъпка стартираме HDFS клъстера, в така наречения single node формат. Можете да погледнете официалната документация тук: http://hadoop.apache.org/common/docs/current/single_node_setup.html . Това може да направите

по два начина:

a) чрез изпълнение на $> bin/start-all.sh

shell командата, като предполагаме, че отново сте позиционирани в hadoop папката.

б) чрез последователно стартиране на различните компоненти на HDFS/MapReduce клсътера, в отделни Bash конзоли:

```
$> bin/hadoop namenode

$> bin/hadoop secondarynamenode

$> bin/hadoop datanode

$> bin/hadoop jobtracker

$> bin/hadoop tasktracker
```

7. Можете да погледнете настройките на така стартирания Hadoop: те са в hadoop0.20.2/conf папката. Файловете core-site.xml, hdfs-site.xml и mapred-site.xml са конфигурирани, точно както е на сайта, даден по-горе.

8. Заедно с Hadoop, е стартиран и Jetty web container, и един прост web application върху него, за администриране и по-скоро мониторинг на Hadoop. Може да видите HDFS UI-а на адрес: http://localhost:50070 , от който може да навигирате по HDFS файловата система и да видите полезна статистика за нея. Може да погледнете и MapReduce UI-а на адрес: http://localhost:50030, който дава доста полезна информация за MapReduce job-овете за вашия Hadoop клъстер.

9. При стартиран Hadoop (!), може да боравите с HDFS от командния ред по следния начин:

```
$> bin/hadoop fs -mkdir Input
```

изпълнена от познатото ви вече място. Може да проверите резултата от вашите операции например през web application-а.

10. Време е да ползваме и Eclipse. Извиняваме се на онези от Вас, които са свикнали с други среди за разработка, и Eclipse им се вижда непозната и нова платформа! Както казахме, ползвайте eclipseEuropa. В подпапка eclipse, стартирайте eclipse изпълнимия файл.

11. IDE-то може да ви попита за Workspace файл - място на файловата система, което съхранява ресурсите от Вашите проекти. Изберете някакво удобно име, което може по-късно да Ви подсети за какво сте го ползвали :)

12. Заедно с Hadoop се дистрибутира и Eclipse plug-in за по-лесна работа и development с продукта (най-често в папка hadoopXX/contrib./eclipse-plugin). Той дефинира нова перспектива: MapReduce.

Изберете MapReduce перспективата: Window -> Open Perspective -> Other -> MapReduce. В долната част на екрана трябва да виждате MapReduce view (view е отновно Eclipse-специфично понятие, като перспективата) с едно жълто слонче като лого. Вътре, в случай че не е конфигуриран Вашия Hadoop, го направете така: от контекстното меню (десен бутон) създайте нов Hadoop location, като въведете следните параметри: за host изберете localhost - изберете го навсякъде, където се иска host информация. Портовете на HDFS и MapReduce компонентите може да ги вземете от конфигурационните xml файлове, които по-горе споменахме. За user изберете например просто user.

13. Ако тази конфигурация е успешна, в PackageExplorer view-то трябва да видите елемент localhost под DFS locations елемента. И ако Hadoop е стартиран (!) трябва да можете да expand-нете това дърво, и да навигирате по вашия HDFS. Може да създавате директории и файлове. Не забравяйте да refresh-нете някой node на дървото, ако не виждате промените си.

---

14. Може да създадете нов MapReduce проект: File -> New -> Other -> MapReduce project, който по същество е обикновен Java проект, но в добавка ще получите доста jar-ове, които автоматично ще се добавят на class path-а ви.

15. Чрез съответен Eclipse wizard, може да създавате следните Hadoop понятия: Mapper, Reducer, Driver. Те представляват Java класове, които предоставят логика, съответна на дадения Hadoop артефакт: логика на map фазата, логика на reduce фазата, а driver-ът е обикновен Java клас, в който се задава конфигурация. В Hadoop MapReduce процесите се наричат job-ове, които, за да се конфигурират, се нуждаят от следната информация: кой е Mapper класа, кой е Reducer класа, от къде да се прочетат входните данни и какъв е техния тип, къде да се запишат изходните данни и какъв е техния тип, и т.н. Тази информация се съдържа в driver класа.

На последното упражнение подкарахме един word count пример, който се оказа 'менте' - всяка дума беше отбелязана с числото 1.

Следващият път ще погледнем този проблем, и ще реализираме 'истински' word count пример върху Hadoop.

16. Можете да shut down-нете Fedora така: изберете log out, и след това ще видите Shut down бутона.

Забележка: Тук целта е по-скоро да се запознаем с API-то за работа с Hadoop и да го видим в действие. Някакви очевидни симптоми на performance gain, за жалост, няма да видим - напротив, един обикновен алгоритъм за word count ще мине далеч по-бързо :) Това е така, защото HDFS дава своите резултати и предимства едва при клъстери с размери от порядъка на стотици машини, всеки node от които съдържа няколко милиона файла, всеки с размер от GB към TB. Вижте hdfs design PDF документа в папките с лекции. В практиката дори не се пишат конкретни MapReduce задачи толкова често, колкото се ползват по-high level езици, които се компилират до няколко MapReduce задачи. Например, върху Hadoop се ползва езика Pig (Yahoo! го ползват много успешно), а върху Google MapReduce се ползва sawzell.

Благодаря Ви за интереса и за посещението :)

Поздрави,

Крум.