

MC851 - Projeto em Computação I
MC855 - Projeto em Sistemas de Computação
MC857 - Projeto em Sistemas de Informação
Explorando o Apache Hadoop

Islene Calciolari Garcia

Instituto de Computação - Unicamp

Segundo Semestre de 2016

Sumário

- 1 Introdução
- 2 Objetivos para MC851/MC855/MC857
- 3 Conceitos básicos
- 4 Experiências anteriores
- 5 Próximos passos

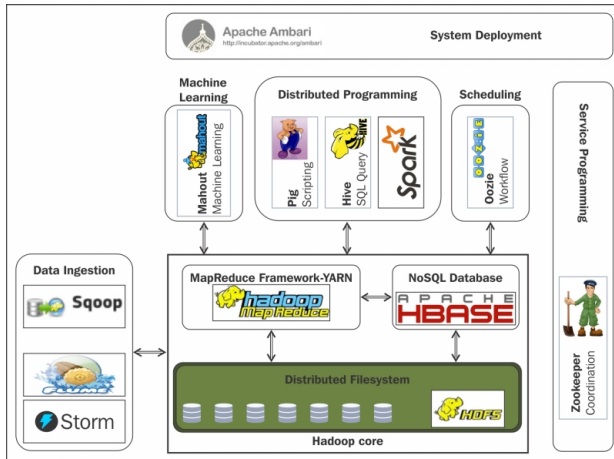
Explorando o Apache Hadoop



- Sistema real! Software livre!
- Big Data *Volume, Velocity, Variety, Veracity*
- Computação distribuída escalável e confiável
- Altamente relevante: usado por empresas como Amazon, Facebook e Yahoo! Veja mais em [Who uses Hadoop?](#)

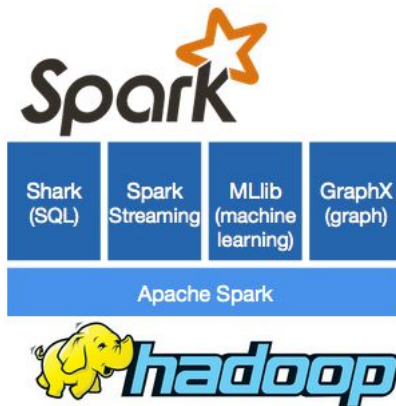
Apache Hadoop Ecosystem

Um ecossistema em evolução



Ref: Hadoop Essentials, Swizec Teller

Spark Stack



Explorando o Apache Hadoop



- Dimensão de usuário:
 - MapReduce
 - Outros sistemas: HBase, ZooKeeper, Hive, Pig
- Dimensão de desenvolvedor:
 - Como contribuir
 - Jira

Critério de avaliação

Projeto 1	13/set ou 15/set
Projeto 2	04/out ou 06/out
Projeto 3	27/out ou 01/nov
Projeto 4	22/nov ou 24/nov
Projeto 5	13/dez ou 15/dez

- 5 projetos com pesos iguais
- Trabalho em grupo (máximo 4 pessoas); disciplina colaborativa
- Todos os alunos deverão apresentar conteúdo nos seminários
- Não haverá exame final

Sobre a escolha dos temas...

- Projetos podem ser:
 - estudo/implementação de aplicações MapReduce ou Spark
 - estudo de um subsistemas do Hadoop (como Pig, Mahout, Zookeeper);
 - análise de issues;
- Pelo menos dois projetos deverão ter experimentação prática;
- Pelo menos dois projetos deverão ter estar na dimensão de desenvolvimento;
- Alunos que estão cursando MC851 ou MC857 e que cursaram MC855 com tema Hadoop poderão apresentar projetos sobre temas já abordados desde que haja um aprofundamento significativo do estudo ou dos resultados obtidos.
- Grupos poderão dar continuidade a projetos sobre um mesmo tema desde que haja um aprofundamento significativo do estudo ou dos resultados obtidos.

Sobre a escolha dos temas...

Sugestão para quem não conhece o Hadoop

Projeto 1	Experimento com MapReduce
Projeto 2	Experimento com Apache Spark
Projeto 3	Análise de <i>issue</i> tipo <i>bug</i>
Projeto 4	Análise de <i>issue</i> tipo <i>umbrella</i>
Projeto 5	Análise de <i>issue</i> livre escolha

Sobre a avaliação dos projetos...

- Os grupos deverão apresentar seminários para a turma e todos os integrantes deverão participar e apresentar conteúdo.
- Pelo menos uma semana antes da apresentação os grupos deverão registrar no Moodle o tema escolhido e um plano de trabalho. Este registro fará parte da nota do projeto.
- Os slides deverão conter referências e informações que permitam avaliar o trabalho desempenhado pelo grupo.
- Todo código desenvolvido deverá ser entregue para avaliação.

Sobre a formação dos grupos...

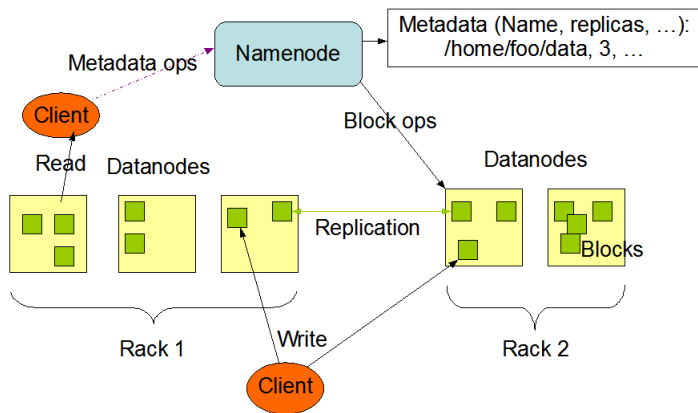
- Grupos são de no máximo 4 alunos;
- Alunos poderão formar grupos com alunos matriculados em outra disciplina ou turma de projetos (MC851B ou MC855AB ou MC857B);
- Alunos que não cursaram outra disciplina de projetos com tema Hadoop poderão formar grupo com alunos que já cursaram.
- Registre o seu grupo nesta [wiki](#)

História do projeto Hadoop

- 2002-2004: Doug Cutting e Mike Cafarella trabalham no projeto Nutch.
 - Nutch deveria indexar a web e permitir buscas
 - Alternativa livre ao Google
- 2003-2004: Google publica artigo sobre o Google File System e MapReduce
- 2004: Doug Cutting adiciona o DFS e MapReduce ao projeto Nutch
- 2006: Doug Cutting começa a trabalhar no Yahoo!
- 2008: Hadoop se torna um projeto Apache
- 2013: Yarn (Hadoop 2)

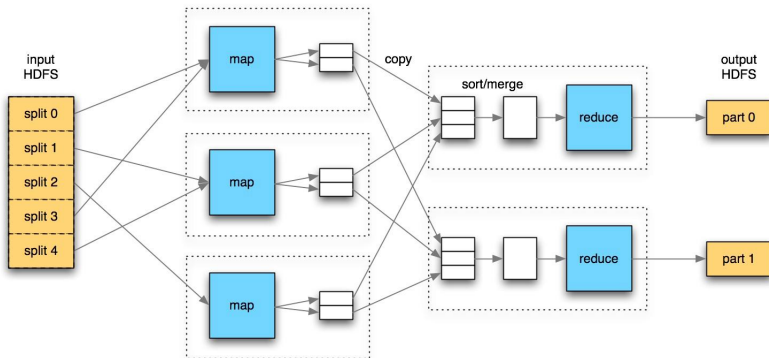
HDFS

HDFS Architecture



Ref: <http://hadoop.apache.org>

MapReduce



Ref: Tom White

Sugestão: experimento com o MapReduce

- Instalar o Hadoop (Single Node Cluster)
- Rodar um exemplo tipo *word count*
- Pesquisar uma aplicação mais elaborada (pode ser do conjunto disponível no diretório de exemplos)
- Explicar abordagem e mostrar código para a turma
- Documentação do experimento (github, bitbucket, moodle)
- Mais detalhes nas próximas aulas.



Sugestão: experimento com o Apache Spark

- Estruturas em memória
- Melhor desempenho em relação ao Hadoop MapReduce
- Pesquisar aplicação e comparar com abordagem anterior
- Explicar abordagem e mostrar código para a turma
- Documentação do experimento (github, bitbucket, moodle)
- Mais detalhes nas aulas após a entrega do primeiro projeto.



Sugestão: Pig, Hive, HBase, Mahout, ...

- Dimensão de usuário
- Como este subsistema se encaixa no ecossistema
- Pesquisar aplicação e comparar com abordagem anterior
- Explicar abordagem e mostrar código para a turma
- Documentação do experimento (github, bitbucket, moodle)
- Sem mais detalhes... :-)



Estudo de entrada no JIRA: Exemplos

- HDFS 6804 race condition between transferring block and appending block causes “Unexpected checksum mismatch exception”
- MAPREDUCE-4584 Umbrella: Preemption and restart of MapReduce tasks
 - Uma entrada marcada como Umbrella envolve vários problemas menores
 - Relacionadas ao desenvolvimento de novas funcionalidades

Análise de entrada no JIRA: Avaliação

- apresentação no laboratório do código e da evolução do grupo com relação ao problema:
 - se for um bug, conseguiram reproduzir?
 - o grupo é capaz de dar alguma sugestão/colaboração?
- apresentação para a turma de seminário com slides
 - descrição do problema
 - resumo das discussões na *issue*
 - resumo dos resultados obtidos

Dicas para escolher

“Professora, você vai dar uma lista com tópicos selecionados?”

- A quanto tempo a issue está aberta?
- Quem reportou o problema ou melhoria?
- A comunidade deu algum retorno?
- Houve discussão?
- Existe possibilidade de contribuição?
- Se a issue estiver fechada? Valeria pelo estudo do processo, mas as chances de contribuição são mínimas...

E se a issue já tiver um *patch* disponível?

- Ótimo!
- O *patch* resolve realmente o problema?
- Está atualizado?
- Estudar o processo de um *patch* também é interessante
- E se um *patch* aparecer durante a análise?
 - o grupo pode escolher outra *issue* (e isso já aconteceu!!!)
 - o grupo pode analisar o *patch* e verificar se teria seguido o mesmo caminho...

Experiência com o ZooKeeper

- Serviço de coordenação
- Turmas: 2s2010, verão-2011, 1s2011, verão-2013
- Inspiração em temas para o Google Summer of Code
- Cluster de tamanho dinâmico
- Receitas



A primeira contribuição! (2011)

- Turma pequena (quase todos tinha cursado no verão)
- Ainda as receitas do ZooKeeper



- Envio de patch

▼  Andre Esteve added a comment - 19/Apr/11 17:16

compareTo.patch aims to correct ordering of ZNodeName objects used to validate lock ownership.

The code at WriteLock gets a list of znodes and for each znode creates a ZNodeName object which is added to a sorted list.

The sorting was based on the full znode name, i.e. x-sessionID-ephemeral_number. As earlier connected clients appear to have lower sessionID values than those which connected latter, who connects first gets the lock disregarding anyone who has already the lock.

This patch simply changes compareTo overload at ZNodeName to just consider the sequence number instead of the full znode name, as this class' objects are used only for this purpose, this seems to have done the trick =)

However, getSessionID not being thread-safe is still an issue.

Could someone try it out and post the results?

[A discussion about this bug and some other issues on lock recipe, as well as this patch contributors, can be found here (in Portuguese) <http://www.lsd.ic.unicamp.br/mc715-1s2011/index.php/Grupo01>]

E o retorno da primeira contribuição?

- Um mês depois...

▼  [Patrick Hunt](#) added a comment - 23/May/11 20:04

Hi, has anyone tried this?

▼  [Matt Abrams](#) added a comment - 30/May/11 23:09

I've tried it. I did see the starvation behavior with the original compareTo method from ZNodeName. When I applied the compareTo patch the starvation issue went away.

- Mais de um ano depois

- Patch seria combinado com outro e incluído
- Grupo comemorou :-)


- O bug continua em aberto...

Kazoo (2013)


- Comunidade ZooKeeper demora para dar retorno?
- Netflix apoiou o desenvolvimento do Curator!
- Kazoo é uma interface em Python para facilitar o uso do ZooKeeper
- Comunidade mais receptiva!
- Aluno de mestrado contribuiu e seu *patch* foi aceito!




E se formos completamente ignorados?

 [Hadoop YARN](#) / [YARN-2299](#)


inconsistency at identifying node

▼  Bruno Alexandre Rosa added a comment - 19/Nov/14 17:33


What are the affected versions?

▼  Bruno Alexandre Rosa added a comment - 19/Nov/14 17:40

Which*

▼  Bruno Alexandre Rosa added a comment - 25/Nov/14 22:24

I tried to reproduce the first case on version 2.5.2 and the bug it is still present. However, instead of host:port1 showing on Lost Nodes, I got host:port2. In the same fashion, I lost track of host:port1. The sum of Lost Nodes remains inconsistent.

▼  Jian He added a comment - 02/May/15 00:26

But when we manager inactive nodes(`RMContextImpl.inactiveNodes`), we use only use host. Two nodes with same host but different port are thought to identical.

This has been fixed in 2.8 to use host:port to track inactive nodes. close this.

A melhor experiência

- Bug simples



Hadoop HDFS / HDFS-6662

WebHDFS cannot open a file if its path contains "%"

- Envio de patch e retorno rápido da comunidade

▼ Gerson Carlos added a comment - 14/Dec/14 13:54

I was able to reproduce the bug on the latest 3.0.0 code from the git repo.

It looks like the uri wasn't being properly encoded before send the request, i.e., % should be converted to %25. So, I added to explorer.js an encoding command.

But after that, the datanode broke when answering, because it wasn't decoding the uris. To fix that, I added to ParameterParser.java the decoding command from QueryStringDecoder.

See the attached hdfs-6662.patch for the diff code.

▼ Haohui Mai added a comment - 15/Dec/14 06:20

```
+ abs_path = encodeURI(abs_path);  
var url = '/webhdfs/v1' + abs_path + '?op=GET_BLOCK_LOCATIONS';
```

I think it should be `encodeURIComponent()` instead of `encodeURI()`.

Can you add a unit test to ensure that the DN decodes the path correctly?

A melhor experiência

- Uma surpresa

Issue Links

breaks

 [HDFS-7816](#) Unable to open webhdfs paths with "+"



CLOSED

- Mais orientações da comunidade e
- final feliz! :-)





Hadoop HDFS / [HDFS-6662](#)

WebHDFS cannot open a file if its path contains "%"

Agile Board

Details

Type:  Bug
Priority:  Critical
Affects Version/s: 2.4.1
Component/s: [namenode](#)
Labels: None

Status: **CLOSED**
Resolution: Fixed
Fix Version/s: [2.7.0](#)

People

Assignee:  Gerson Carlos

Reporter:  Brahma Reddy Battula

Outras experiências...

- A issue mais simples: [HDFS-9121](#)
- Um bug já resolvido [HDFS-8312?](#)
- Às vezes a comunidade dá bons conselhos:
[MAPREDUCE-6487](#)
- Às vezes nem tanto... [MAPREDUCE-6467](#)

Ainda sobre a escolha das issues

- Contribuir para o projeto Hadoop não é o objetivo principal da disciplina...
- Escolham temas adequados ao perfil de cada grupo:
 - alguns alunos escolhem temas relacionados ao trabalho/pesquisa
 - alguns grupos gostam de temas complexos (ex. Erasure Coding)
 - outros gostam de explorar problemas sem solução (Won't fix!)
- Happy Hacking!

Próxima aula

- Dia 25, na sala 85
- Tema: HDFS e Mapreduce
- Primeiro experimento com Hadoop

Aulas 30/ago e 1, 6, 8/set

- Atendimento sala 302

Referências

- Site do Apache Hadoop: <http://hadoop.apache.org/>
- Hadoop The Definitive Guide, 4th Edition, Tom White