AVALIANDO AS MÉTRICAS DE QUALIDADE DO PROJETO DA API MAPREDUCE DO HADOOP USANDO ANÁLISE POR ENVOLTÓRIA DE DADOS

Andre de O. Kovacs
Programa de Mestrado em Pesquisa Operacional, Faculdade de Ciências Aplicadas, Universidade Estadual de Campinas
R. Pedro Zaccaria, 1300 - Caixa Postal 1068, CEP 13484-350 - Limeira - São Paulo
andre.kovacs@fca.unicamp.br

RESUMO

O *software* Hadoop disponibiliza duas versões incompatíveis entre si da sua API *MapReduce* por conta de seguirem diferentes padrões de projeto e terem passado por uma atualização recente para a migração para a nova plataforma YARN. Porém, a avaliação do impacto de tais fatores na qualidade do seu projeto do *software* é uma tarefa complexa devido a relações de compromisso entre diferentes aspectos, tais como complexidade, coesão, capacidade de reuso, estabilidade e abstração do código fonte. Este estudo examina os efeitos nas métricas de qualidade de *software* da API MapReduce do Hadoop pela análise estática do código fonte e Análise por Envoltória de Dados (DEA). A análise inclui testes de significância de mudanças de indicadores de eficiência DEA como resultado de diferentes combinações dos fatores supracitados, além da comparação dos indicadores de eficiência DEA com as métricas de índices de produtividade correspondentes e inclui testes das distribuições de probabilidade dos mesmos.

PALAVARAS CHAVE: Análise por Envoltória de Dados, Interface de Programação de Aplicação, Análise Estática de Código.

Área principal: TEL&SI - PO em Telecomunicações e Sistemas de Informações, DEA - Análise Envoltória de Dados, ADM - Apoio à Decisão Multicritério.

ABSTRACT

Hadoop software provides two incompatible versions of its MapReduce API, using distinct design patterns, and it was recently updated to migrate to the new YARN platform. However, the evaluation of the impact of such factors on its software design quality is a complex task due to trade-offs between different aspects, such as source code complexity, cohesion, reusability, stability, and abstraction. This study examines the effects on software quality metrics of Hadoop MapReduce API by static code analysis and Data Envelopment Analysis (DEA). The analyses include: significance testing of DEA performance rating changes, as a result of different combinations of the factors above; a comparison of DEA performance ratings to that of corresponding productivity ratios; and probability distribution tests of the data.

KEYWORDS. Data Envelopment Analysis. Application Programming Interface. Static Code Analysis.

Main area: TEL&SI – OR in Telecommunications and Information Systems, DEA – Data Evelopment Analysis, ADM - Multicriteria Decision Support.

1. Introdução

O *software* Hadoop, desde a sua disponibilização ao público em geral, no final de 2011 (WHITE, 2012), introduziu uma nova versão da sua API MapReduce para a criação de aplicações utilizando a plataforma (disponível no pacote org.apache.hadoop.mapreduce), buscando facilitar a sua evolução futura. Essa nova versão é incompatível com a versão antiga (disponível no pacote org.apache.hadoop.mapred). No entanto, inicialmente, por motivos de compatibilidade com aplicações e códigos de exemplo pré-existentes, a versão antiga da API foi mantida ativa e, desde então, permanece disponível e atualizada. Além disso, o *software* Hadoop passou recentemente por uma migração de arquitetura de *software* para a sua compatibilização com a plataforma YARN.

Este estudo buscou avaliar a qualidade da API MapReduce do *software* Hadoop, dentre as diferentes versões do mesmo, considerando como possíveis fatores impactantes os diferentes padrões de projeto utilizados e a mudança na arquitetura do *software* com a migração para a plataforma YARN. Realizou-se um estudo quantitativo longitudinal de métricas de qualidade de *software* referentes à análise estática do projeto (MARTIN, 2003) das APIs MapReduce nova e antiga do *software* Hadoop.

Modelos estáticos em UML foram obtidos pela engenharia reversa do código fonte do *software*, através da ferramenta CodeMiner, para as atualizações mais recentes das diferentes linhas de versões e suas respectivas versões de base, desde a primeira versão estável disponibilizada ao público em geral. Em seguida, diferentes métricas de qualidade de *software* foram coletadas, utilizando a ferramenta SDMetrics, de forma agregada por pacotes. E, por fim, os dados referentes às métricas selecionadas foram usados para a análise comparativa da eficiência do projeto das APIs MapReduce para diferentes agrupamentos das versões do Hadoop. Para tanto, utilizaram-se a técnica de Análise por Envoltória de Dados (DEA, do inglês *Data Envelopment Analysis*) (ZHU, 2009) e testes estatísticos de significância dos seus resultados (BANKER e NATARAJAN, 2011), de forma a comparar as eficiências dentre as versões do *software* Hadoop. Já para avaliar o comportamento da técnica DEA no problema, utilizou-se uma análise comparativa da mesma com a técnica baseada em índices de produtividade, tradicional e comumente utilizada na literatura para esse fim.

Este artigo inicia contextualizando o histórico e as diferenças nos padrões de projeto utilizados na API MapReduce do *software* Hadoop. Na sequência, é apresentada uma revisão da literatura referente ao uso da DEA em problemas de Engenharia, e em especial em Engenharia de *Software*. Em seguida, é detalhada a metodologia do estudo. Por fim, são apresentados e discutidos os resultados e as conclusões obtidas.

. . drões de r. dêõ d. A I M. pReduce

A API MapReduce do *software* Hadoop foi inicialmente desenvolvida seguindo um padrão de projeto de *software* baseado no uso de interfaces e disponibilizada no pacote org.apache.hadoop.mapred. Posteriormente, com o lançamento da versão 0.20.0 do Hadoop, uma nova versão da API MapReduce foi introduzida com o intuito de substituir a versão antiga da API, que é baseada no padrão de projeto *context object* e no uso extensivo de classes abstratas e disponibilizada no pacote org.apache.hadoop.mapreduce. Porém, apenas com o lançamento da versão 0.22.0 do Hadoop é que o nível de funcionalidades de ambas APIs foi equiparado. Tentativas pelo time de desenvolvimento do Hadoop de descontinuar a versão antiga da API nos lançamentos iniciais da versão 0.20.x foram frustradas, retornando a API antiga em lançamentos posteriores, de forma que hoje ambas APIs encontram-se ativamente sendo atualizadas com novas funcionalidades e disponíveis para uso (WHITE, 2012).

Na linguagem de programação Java, utilizada no desenvolvimento do código fonte do Hadoop, interfaces e classes abstratas são utilizadas na prática como um mecanismo de definição do contrato da API, que consiste no conjunto de requisições às quais ela pode responder. Entretanto, classes abstratas também são capazes de promover o reuso de funcionalidade por meio da herança de classes existentes e permitir a definição de famílias de objetos com interfaces idênticas (GAMMA, HELM, *et al.*, 1994), sendo, assim, potencialmente mais flexíveis do que

interfaces. O uso do padrão *context object* é considerado como uma boa prática no projeto de APIs de *software* para que elas possam ser estendidas transparentemente para a adição de funcionalidades não-padrão pelos usuários, alterando o seu comportamento de forma dinâmica, sem a necessidade alteração do código fonte padrão da API (SCHMIDT, STAL, *et al.*, 2004).

3. Análise de Eficiência: Revisão da Literatura

A Análise por Envoltória de Dados (DEA, do inglês *Data Envelopment Analysis*) é um método de Análise Multicritério baseado em uma abordagem não paramétrica, por ser orientada aos dados, capaz de avaliar a eficiência da performance de um conjunto de entidades denominadas *Decision-Making Units* (DMUs). A DEA permite a análise de relações entre as DMUs, muitas vezes complexas e desconhecidas, entre múltiplas variáveis de insumo (do inglês *input*) e variáveis de produto (do inglês *output*) para a definição da fronteira empírica de eficiência ou de máxima produtividade. Ela é executada por técnicas de otimização matemática, normalmente utilizando Programação Linear (ZHU, 2009). Desta forma, uma DMU está na fronteira eficiente com base na evidência disponível se e somente se a performance de outras DMUs não evidenciam que algum de seus insumos ou produtos pode ser melhorado sem prejudicar algum dos insumos ou produtos das outras, *i.e.*, a DMU em questão não é dominada por nenhuma outra.

Além disso, a sua modelagem pode ser orientada aos insumos ou aos produtos. No primeiro caso, busca-se determinar o nível de eficiência de cada DMU pela distância do nível atual ao ideal dos produtos para o nível atual dos insumos, sendo assim denominada como formulação orientada aos insumos. Da mesma forma, se considerarmos as múltiplas razões dos insumos pelos produtos para cada DMU, pela formulação orientada aos produtos, teremos a determinação do nível de ineficiência de cada DMU, ou seja, o inverso da eficiência, pela distância do nível atual ao ideal dos insumos para o nível atual dos produtos.

Em contextos onde a variação da métrica de eficiência é desproporcional às reduções dos insumos ou elevação dos produtos, a variante do modelo de DEA não-relacional proposto por Fare e Lovell deve ser utilizado em lugar do modelo por envoltória proposto por Charnes *et al* (ZHU, 2009). Já os fundamentos de diferentes estimadores estatísticos das medidas de ineficiência obtidas pelo DEA, que correspondem ao inverso das medidas de eficiência, e testes estatísticos adequados para a comparação de eficiência entre dois grupos de DMUs foram desenvolvidos por Banker (BANKER e NATARAJAN, 2011).

Conforme apresentado por Charnes e Cooper (CHARNES e COOPER, 1985), DEA é um procedimento de análise de eficiência vantajoso, pois é capaz de incorporar métricas conflitantes em uma medida única, permite análises estatísticas subsequentes e não necessita de um valor de referência para a análise. Porém, o procedimento assume que a fronteira eficiente empírica pode ser aproximada por uma combinação linear por partes das observações mais eficientes.

Avaliações de eficiência de performance utilizando DEA são, na sua maioria, focadas em Economia. Entretanto, aplicações de DEA em Engenharia são abordadas por Triantis (TRIANTIS, 2004), especificamente para a avaliação de alternativas de projetos. Ele conjectura que ela poderia ser utilizada na busca por de configurações do projeto de produtos. A aplicação pioneira de DEA em Engenharia de *Software* foi proposta por Banker e Kemerer, em 1989 (BANKER e KEMERER, 1989). Nela, comparam o uso de análise de regressão contra DEA na avaliação de ganhos de escala em projetos de desenvolvimento de novos *softwares*, considerando como insumo o volume de homens-hora dedicados ao projeto e como produto o tamanho do *software* desenvolvido. Posteriormente, Paradi *et al*. (PARADI, REESE e ROSEN, 1997) avaliaram a produtividade em projetos de implantação de ERP para a previsão de esforço para futuros projetos utilizando DEA e análise por regressão. Chatzoglou e Soteriou (CHATZOGLOU e SOTERIOU, 1999) conduziram estudos comparativos dos abordagens de avaliação de produtividade pelo método baseado em indicadores de produtividade tradicional *versus* DEA e a investigação dos efeitos de qualidade em projetos de melhoria de *softwares* para dois grandes bancos canadenses. E Parthasarathy e Anbazhagan (PARTHASARATHY e ANBAZHAGAN,

2008) avaliaram a eficiência do processo de Captura e Análise de Requisitos (RCA) no desenvolvimento de *software* utilizando DEA.

Os estudos de Chatzoglou e Soteriou são especialmente relevantes, pois demonstram que: a abordagem por DEA é capaz de suprir deficiências do método baseado em indicadores de produtividade; a qualidade pode impactar significantemente na eficiência e custo de projetos de *software*; e resultados enganadores podem decorrer do uso da definição radial de produtividade. Contudo, eles consideram como insumo o custo do projeto de desenvolvimento e como produtos: a medida de qualidade pela "porcentagem de retrabalho" (relação do volume de horas de retrabalho dividido pelo total de horas de desenvolvimento); o tamanho do *software* dado em pontos de função; e a duração do projeto. Desta forma, tratam da qualidade do ponto de vista das métricas do projeto de desenvolvimento e não das métricas da qualidade do projeto do *software* em si.

Tradicionalmente, a avaliação de projetos de desenvolvimento de *software* é baseada no uso de diversos índices de produtividade unidimensionais, usualmente para produtividade e qualidade. Porém, pela simplicidade e independência de tais métricas, dificilmente é possível estimar precisamente os benefícios da melhoria da qualidade do *software* (CHATZOGLOU e SOTERIOU, 1999). Geralmente, é utilizada, na análise desses dados, a propriedade estatística de que índices de produtividade apresentam uma distribuição normal quando há proporcionalidade estrita do numerador pelo denominador, não suportando um termo de intercepto, um termo de intermédio ou uma relação não-linear dos mesmos.

Em avaliações do projeto de *software* baseados em Orientação à Objetos (OO), é possível utilizar diretamente métricas do tamanho e qualidade *software*, analisando o seu projeto a partir do seu código fonte, em diversos níveis e visões distintas. Usualmente, são avaliados apenas aspectos estáticos ou dinâmicos do *software*, através da representação do *software* em modelos UML. Entretanto, tradicionalmente, as métricas são analisadas individualmente e descritivamente no tempo ou de forma agregada, havendo exceção apenas para a análise de relações de compromisso entre as métricas de abstração e instabilidade, seguindo o princípio de abstrações estáveis (SAP), em que são analisadas as dispersões dos pontos em quadrantes e os números de desvios em relação à sequência principal (MARTIN, 2003).

4. Metodologia do Estudo

Foram utilizados, neste estudo, dados de métricas de qualidade de *software* das versões antiga e nova da API MapReduce, considerando as atualizações mais recentes das diferentes linhas de versões e suas respectivas versões de base do *software* Hadoop, desde a primeira versão estável disponibilizada ao público em geral até o final do ano de 2013. As medidas foram obtidas por meio da geração de modelos estáticos do código fonte em UML pela ferramenta CodeMiner, após ajustes no *script* de coleta de dados para complementação dos dados e da posterior análise descritiva dos modelos UML pela ferramenta SDMetrics. De forma a abranger o maior número de aspectos de qualidade do projeto do *software* possível, selecionaram-se as métricas no nível de pacote (MARTIN, 2003), que consistem em: número total de operações em classes e interfaces (NumOps_tc**)**; coesão relacional (H); número de pacotes aos quais classes e interfaces dependem (DepPack); número de relações entre classes e interfaces (R); e distância normalizada para a sequência principal (ND). Para a agregação final das métricas para os diferentes pacotes principais compreendendo as duas APIs, adotaram-se: o número total de operações normalizado, dado pela variável NNumOps_tc definida pela razão entre NumOps_tc e o máximo de NumOps_tc, como medida do tamanho; a média de H como medida de coesão; a média de DepPack como medida de acoplamento; a soma de R como medida de complexidade; e a soma de ND como medida para SAP (abstração *versus* instabilidade).

Posteriormente, as métricas agregadas de qualidade do projeto do *software* foram classificadas em insumos e produtos de forma que o procedimento de análise de dados DEA pudesse produzir o indicador de eficiência para cada combinação das versões de API MapReduce e Hadoop. Visto que o número total de operações dos objetos é dependente do nível de funcionalidade provido pela API e não do padrão de projeto do *software*, essa métrica foi adotada

como um insumo no DEA. Em contrapartida, dado que as demais métricas são dependentes do projeto de *software*, elas foram adotadas como produtos no DEA. Adotou-se o modelo DEA não-relacional baseado nos insumos e com ganhos de escala constantes proposto por Fare e Lovell (ZHU, 2009) por permitir aumentos desproporcionais nos produtos para um mesmo nível de insumos. Entretanto, pelo fato de o modelo DEA adotado assumir que os insumos e produtos devem ser maximizados, utilizou-se o inverso das métricas agregadas NNumOps_tc, DepPack, R e ND, já que em um projeto eficiente de *software* estas métricas devem ser minimizadas (MARTIN, 2003). Segue-se a premissa de modelos de produtividade de que se deseja maximizar os produtos, mas mantendo o nível atual dos insumos.

Em seguida, os indicadores de eficiência das DMUs foram agrupados por API e versão principal do Hadoop, de forma a testar a primeira hipótese nula, utilizando o teste de Banker para a comparação de grupos de DMUs: (H0,1) - não há diferença estatisticamente significativa na eficiência das métricas de qualidade do projeto entre diferentes agrupamentos de versões das APIs MapReduce.

Buscando analisar a relação entre os insumos e produtos, realizaram-se também a análise de regressão e o teste estatístico de Shapiro-Wilk, de forma a testar as terceira e quarta hipóteses nulas: (H0,2) - os índices de produtividade das métricas de qualidade de *software* selecionadas são normalmente distribuídas; (H0,3) - os coeficientes do modelo de regressão linear não contribuem de forma significativa para a previsão do valor de saída; (H0,4) - os indicadores de eficiência DEA do projeto das APIs MapReduce são normalmente distribuídos.

No entanto, o teste de Banker para os indicadores de eficiência do DEA varia conforme a distribuição dos mesmos. Logo, utilizou-se o teste de Kolmogorov-Smirnof, de forma a testar as duas últimas hipóteses nulas: (H0,5) - os indicadores de ineficiência do projeto das APIs MapReduce são exponencialmente distribuídos; (H0,6) - os indicadores de ineficiência do projeto das APIs MapReduce seguem uma distribuição meia-normal.

Os testes de significância foram executados utilizando o nível de significância padrão de 0,05 para a análise do Valor-p, de forma a avaliar as hipóteses aqui propostas. Porém, dado que múltiplas comparações de uma mesma DMU foram necessárias na aplicação do teste de Banker, com 6 no máximo, um nível de significância do teste de 0,0083 foi utilizado, decorrente da aplicação do método de correção de Bonferroni para testes com múltiplas comparações.

5. Resultados

A relação de versões analisadas do *software* Hadoop e das métricas de qualidade utilizadas, com seus respectivos valores coletados, encontram-se listados na Tabela 1:

Versão	Data de lançamento	API	NNumOps_tc (Tamanho)	ND (SAP)	DepPack (Acopla-mento)	R (Complexidade)	H (Coesão)
1.0.4	04-10-2012	org.apache.hadoop.mapred	0,748	1,358	0,875	126,000	0,456
		org.apache.hadoop.mapreduce	0,187	1,125	0,500	49,000	0,399
1.1.2	31-01-2013	org.apache.hadoop.mapred	0,755	1,359	0,875	148,000	0,475
		org.apache.hadoop.mapreduce	0,188	1,125	0,500	48,000	0,397
1.2.1	22-07-2013	org.apache.hadoop.mapred	0,763	1,242	3,625	148,000	0,475
		org.apache.hadoop.mapreduce	0,189	1,208	1,813	56,000	0,413
0.22.0	04-12-2011	org.apache.hadoop.mapred	0,640	1,124	2,500	128,000	0,464
		org.apache.hadoop.mapreduce	0,490	2,285	0,720	97,000	0,388
2.0.0	16-05-2012	org.apache.hadoop.mapred	0,481	1,067	4,000	133,000	0,429
		org.apache.hadoop.mapreduce	0,975	10,285	0,808	103,000	0,333
0.23.9	01-07-2013	org.apache.hadoop.mapred	0,453	1,063	4,143	98,000	0,379
		org.apache.hadoop.mapreduce	0,151	11,359	0,846	103,000	0,403
2.1.1	15-08-2013	org.apache.hadoop.mapred	0,479	1,146	3,429	103,000	0,381
		org.apache.hadoop.mapreduce	0,994	12,379	0,778	108,000	0,430
2.2.0	07-10-2013	org.apache.hadoop.mapred	0,481	1,061	6,429	120,000	0,406
		org.apache.hadoop.mapreduce	1,000	10,682	2,036	121,000	0,438

Tabela 1. Valores das métricas para cada versão do Hadoop e API

Primeiramente, foram validadas as premissas para o uso dos índices de produtividade aqui propostos, para a análise de regressão linear e testes de significância para a normalidade dos dados das mesmas e os seus dados estão resumidos nas Tabela 2 e Tabela 3. O primeiro tipo de dados busca validar a hipótese H0,3 e estão agrupados por modelo de regressão linear. Já o segundo tipo de dados busca validar a hipótese H0,2 e está detalhado por versão da API MapReduce. Os casos em que a hipótese nula é rejeitada ou a qualidade do modelo linear é adequada encontram-se destacados em negrito:

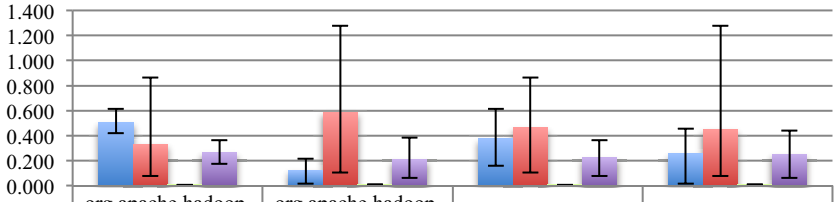
Variável dependente	Variável independente	Coeficiente		Valor-p		Razão F	Coeficiente de determinação (R^2) [%]
		Inclinação	Intercepto	Inclinação	Intercepto		
1/ND	1/NNumOps_tc	0,021	**0,587**	0,678	**0,002**	0,180	1,272
1/DepPack		0,140	0,512	0,101	0,064	**3,072**	17,996
1/R		**0,002**	**0,006**	**0,001**	**0,001**	**17,633**	**55,742**
H		-0,005	**0,431**	0,330	**0,000**	1,021	6,794

Tabela 2. Análise de regressão das métricas de qualidade pelo tamanho da API (hipótese H0,3)

API	(1/ND)/ (1/NNumOps_tc)	(1/DepPack)/ (1/NNumOps_tc)	(1/R)/ (1/NNumOps_tc)	H/ (1/NNumOps_tc)	Eficiência
org.apache.hadoop.mapred	0,182	**0,003**	0,869	0,063	**0,018**
org.apache.hadoop.mapreduce	0,690	0,174	0,253	**0,032**	0,186

Tabela 3. Valores p dos testes de Shapiro-Wilk para a normalidade das razões das relações de produtividade (hipótese H0,2) e indicador de eficiência da DEA (hipótese H0,4) agrupados por APIs

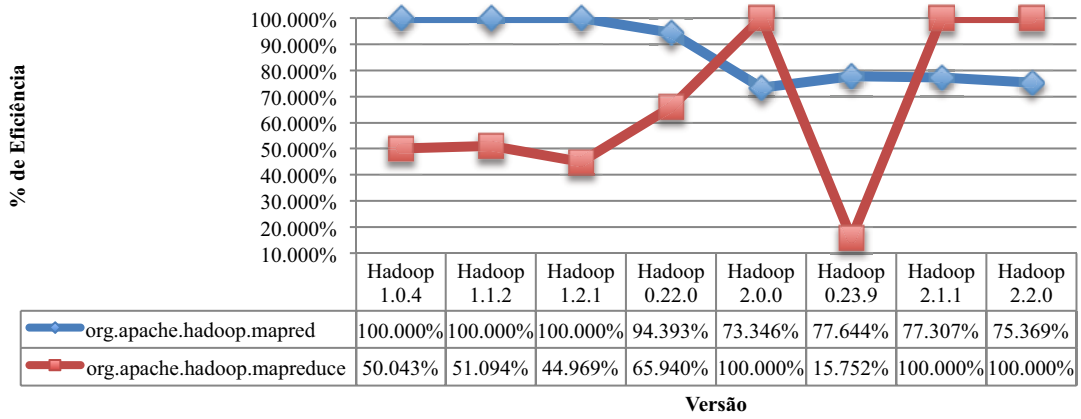
Em seguida, foi realizada uma análise descritiva da dispersão dos dados das métricas de qualidade por índices de produtividade, agrupados por versão da API MapReduce ou versão do Hadoop. Obtiveram-se os dados da Figura 1:



	org.apache.hadoop.mapred	org.apache.hadoop.mapreduce	Hadoop 1.x	Hadoop 2.x
(1/ND)/(1/NNumOps_tc)	0.505	0.123	0.374	0.254
(1/DepPack)/(1/NNumOps_tc)	0.329	0.586	0.465	0.450
(1/R)/(1/NNumOps_tc_)	0.005	0.006	0.005	0.006
H/(1/NNumOps_tc)	0.264	0.208	0.222	0.251

Figura 1. Análise da dispersão das métricas de qualidade por índices de produtividade

Como os dados da Tabela 2 evidenciaram que os índices de produtividade adotados não apresentam comportamento linear. Partiu-se então para a análise por DEA utilizando o modelo não-relacional de Fare e Lovell. Os dados dos indicadores de eficiência desta análise encontram-se sintetizados na Figura 2:



	Hadoop 1.0.4	Hadoop 1.1.2	Hadoop 1.2.1	Hadoop 0.22.0	Hadoop 2.0.0	Hadoop 0.23.9	Hadoop 2.1.1	Hadoop 2.2.0
org.apache.hadoop.mapred	100.000%	100.000%	100.000%	94.393%	73.346%	77.644%	77.307%	75.369%
org.apache.hadoop.mapreduce	50.043%	51.094%	44.969%	65.940%	100.000%	15.752%	100.000%	100.000%

Figura 2. Medidas do indicador de eficiência DEA das APIs pela versão relativa do Hadoop

Posteriormente, de forma a validar a adequação do comportamento dos modelos da DEA, realizou-se a comparação por correlação das medidas dos índices de produtividade com os indicadores de eficiência da DEA para os modelos dos produtos individualmente e conjuntamente, resultando nas Tabela 4 e Tabela 5:

Versão	API	(1/ND)/ (1/NNumOps_tc)		(1/DepPack)/ (1/NnumOps_tc)		(1/R)/ (1/NnumOps_tc)		H/ (1/NnumOps_tc)	
		Relação	DEA	Relação	DEA	Relação	DEA	Relação	DEA
1.0.4	org.apache.hadoop.mapred	0,551	0,897	0,855	0,670	0,006	0,627	0,341	0,778
	org.apache.hadoop.mapreduce	0,166	0,270	0,373	0,293	0,004	0,403	0,074	0,170
1.1.2	org.apache.hadoop.mapred	0,556	0,904	0,863	0,675	0,005	0,539	0,359	0,818
	org.apache.hadoop.mapreduce	0,167	0,272	0,376	0,294	0,004	0,414	0,075	0,170
1.2.1	org.apache.hadoop.mapred	0,615	1,000	0,211	0,165	0,005	0,545	0,362	0,827
	org.apache.hadoop.mapreduce	0,157	0,255	0,104	0,082	0,003	0,357	0,078	0,178
0.22.0	org.apache.hadoop.mapred	0,570	0,926	0,256	0,200	0,005	0,528	0,297	0,677
	org.apache.hadoop.mapreduce	0,214	0,349	0,680	0,533	0,005	0,534	0,190	0,433
2.0.0	org.apache.hadoop.mapred	0,451	0,733	0,120	0,094	0,003	0,382	0,206	0,470
	org.apache.hadoop.mapreduce	0,094	0,153	1,207	0,945	0,009	1,000	0,324	0,740
0.23.9	org.apache.hadoop.mapred	0,426	0,693	0,109	0,086	0,005	0,488	0,172	0,392
	org.apache.hadoop.mapreduce	0,013	0,022	0,179	0,140	0,001	0,155	0,061	0,139
2.1.1	org.apache.hadoop.mapred	0,418	0,680	0,140	0,109	0,005	0,491	0,183	0,416
	org.apache.hadoop.mapreduce	0,080	0,130	1,278	1,000	0,009	0,972	0,427	0,974
2.2.0	org.apache.hadoop.mapred	0,453	0,737	0,075	0,059	0,004	0,423	0,195	0,446
	org.apache.hadoop.mapreduce	0,094	0,152	0,491	0,384	0,008	0,873	0,438	1,000
	Correlação	**0,999**		**0,999**		**0,999**		**1,000**	

Tabela 4. Comparação dos indicador de eficiência unidimensionais por DEA versus por índices de produtividade individuais

CV	(1/ND)/ (1/NNumOps_tc)	(1/DepPack)/ (1/NNumOps_tc)	(1/R)/ (1/NNumOps_tc)	H/ (1/NNumOps_tc)
Eficiência	0,483	0,508	0,772	0,914

Tabela 5. Coeficientes de correlação entre as medidas das índices de produtividade versus o indicador de eficiência do modelo DEA combinado

Adicionalmente, por meio da DEA, obtiveram-se também quais seriam os valores das métricas de qualidade se todas DMUs fossem projetadas na fronteira eficiente empírica. Como base comparativa, a análise de regressão linear dos índices de produtividade foi realizada para esses dados. Seus resultados encontram-se na Tabela 6. Os dados buscam validar a hipótese H0,3 e estão agrupados por modelo de regressão linear. Os casos em que a hipótese nula é rejeitada ou a qualidade do modelo linear é adequada encontram-se destacados em negrito:

Variável dependente	Variável independente	Coeficiente		Valor-p		Razão F	Coeficiente de determinação (R^2) [%]
		Inclinação	Intercepto	Inclinação	Intercepto		
1/ND	1/NNumOps_tc	0,054	**0,527**	0,226	**0,002**	1,603	10,276
1/DepPack		**1,349**	**-0,703**	**0,000**	**0,000**	**815,094**	**98,311**
1/R		**0,009**	**-0,004**	**0,000**	**0,002**	**688,145**	**98,006**
H		**0,433**	**-0,089**	**0,000**	**0,000**	**5284,234**	**99,736**

Tabela 6. Análise de regressão das métricas de qualidade pelo tamanho da API, para a projeção das DMUs na fronteira eficiente

De forma a avaliar a distribuição dos dados dos indicadores de ineficiência (inverso da eficiência) do modelo DEA combinado, para a validação, como sugerido por Banker, da aderência dos mesmos às distribuições exponencial e meia-normal, utilizou-se o teste estatístico de Kolmogorov-Smirnov para agrupamentos por versão da API MapReduce e versão do *software* Hadoop dos indicadores de ineficiência de DMUs, onde obteve-se os dados da Tabela 7. Os casos onde as hipóteses nulas são rejeitadas encontram-se destacadas em negrito:

Distribuição	Variável	Agrupamento		Valor p
		API	Versão	
Exponencial em [0,∞)	ln(1/Eficiência)	org.apache.hadoop.mapred	Hadoop 1.0.4 a 2.2.0	0,211
		org.apache.hadoop.mapreduce	Hadoop 1.0.4 a 2.2.0	0,087
Meia-normal em [0,∞)	ln(1/Eficiência)	org.apache.hadoop.mapred	Hadoop 1.0.4 a 2.2.0	0,270
		org.apache.hadoop.mapreduce	Hadoop 1.0.4 a 2.2.0	0,088
Exponencial em [0,∞)	ln(1/Eficiência)	Ambas	Hadoop 1.0.4 a 0.22.0	0,574
		Ambas	Hadoop 2.0.0 a 2.2.0	0,505
Meia-normal em [0,∞)	ln(1/Eficiência)	Ambas	Hadoop 1.0.4 a 0.22.0	0,270
		Ambas	Hadoop 2.0.0 a 2.2.0	**0,004**
Exponencial em [0,∞)	ln(1/Eficiência)	org.apache.hadoop.mapred	Hadoop 1.0.4 a 0.22.0	**0,022**
		org.apache.hadoop.mapreduce	Hadoop 2.0.0 a 2.2.0	**0,022**
Meia-normal em [0,∞)	ln(1/Eficiência)	org.apache.hadoop.mapred	Hadoop 1.0.4 a 0.22.0	0,211
		org.apache.hadoop.mapreduce	Hadoop 2.0.0 a 2.2.0	0,211
Exponencial em [0,∞)	ln(1/Eficiência)	org.apache.hadoop.mapred	Hadoop 2.0.0 a 2.2.0	0,112
		org.apache.hadoop.mapreduce	Hadoop 1.0.4 a 0.22.0	0,326
Meia-normal em [0,∞)	ln(1/Eficiência)	org.apache.hadoop.mapred	Hadoop 2.0.0 a 2.2.0	0,699
		org.apache.hadoop.mapreduce	Hadoop 1.0.4 a 0.22.0	0,211

Tabela 7. Testes de Kolmogorov-Smirnof para a distribuição dos indicadores de ineficiência (hipótese H0,5 e H0,6) agrupados por API e versão

Por fim, dada a existência de DMUs para ambas versões das APIs MapReduce pertencentes à fronteira eficiente empírica, foram realizados múltiplos testes estatísticos de significância dos dados dos indicadores de ineficiência (inverso da eficiência) do modelo DEA combinado, de forma a analisar comparativamente a eficiência das DMUs. Como sugerido por Banker, os indicadores de ineficiência foram agrupados por versão da API MapReduce e versão do *software* Hadoop para a validação comparativa entre os grupos. Os dados encontram-se na Tabela 8. Utilizou-se um nível de significância do teste de 0,0083, decorrente da aplicação do método de correção de Bonferroni para testes com 6 comparações. Os casos em que a hipótese nula é rejeitada encontram-se destacados em negrito:

Distribuição	Grupo 1	Grupo 2	Estimador	Valor p
Exponencial	org.apache.hadoop.mapreduce	org.apache.hadoop.mapred	16,658	**0,005**
Exponencial	Hadoop 2.0.0 a 2.2.0	Hadoop 1.0.4 a 0.22.0	1,119	0,825 (bicaudal)
Meia-normal	org.apache.hadoop.mapreduce Hadoop 2.0.0 a 2.2.0	org.apache.hadoop.mapred Hadoop 1.0.4 a 0.22.0	32,031	**0,003**
Meia-normal	org.apache.hadoop.mapreduce Hadoop 1.0.4 a 0.22.0	org.apache.hadoop.mapred Hadoop 2.0.0 a 2.2.0	0,428	0,431 (bicaudal)
Meia-normal	org.apache.hadoop.mapred Hadoop 2.0.0 a 2.2.0	org.apache.hadoop.mapred Hadoop 1.0.4 a 0.22.0	19,119	**0,007**
Meia-normal	org.apache.hadoop.mapreduce Hadoop 1.0.4 a 0.22.0	org.apache.hadoop.mapreduce Hadoop 2.0.0 a 2.2.0	0,717	0,755 (bicaudal)

Tabela 8. Testes de Banker monocaudais (hipótese H1,1 com Grupo 1 > Grupo 2) para agrupamentos dos indicadores de ineficiência das APIs (hipótese H0,1 com Grupo 1 = Grupo 2)

5. Discussão

Análises estatísticas comparativas das relações de produtividade utilizadas neste estudo contra os indicadores de eficiência dos modelos DEA com produtos univariados mostrou que o comportamento do indicador de eficiência do modelo DEA aqui proposto é condizente com o obtido através de relações de produtividade, dado que coeficientes de correlação obtidos foram todos entre 0,999 e 1,000. Além disso, a análise de correlação entre os indicadores de eficiência deduzidos pelo modelo DEA combinando todos os produtos e as relações de produtividade individuais indicou um grande efeito de todas as métricas aqui consideradas sobre os indicadores de eficiência, justificando as suas utilizações no modelo DEA combinado.

Evidenciaram-se casos que inviabilizam técnicas estatísticas convencionais requerendo dados normalmente distribuídos, baseado nos resultados do teste de normalidade de Shapiro-Wilk para a adequação do indicador de eficiência do modelo DEA combinado e das relações de produtividade utilizadas neste estudo à premissa de normalidade univariada.

Ademais, a análise de dispersão das métricas das relações de produtividade evidenciou uma relação de compromisso nos padrões de projeto das APIs para a métrica ND em detrimento da métrica DepPack com a API antiga privilegiando o princípio de abstrações estáveis (SAP) e a nova API privilegiando a capacidade de reuso (baixo acoplamento).

O resultado da análise de regressão evidenciou que o modelo linear mostrou-se adequado para as relações de produtividade baseadas nas métricas de coesão relacional (H), número de pacotes aos quais classes e interfaces dependem (DepPack) e número de relações entre classes e interfaces (R). Pela análise de regressão, evidenciou-se também que, exceto para número de pacotes aos quais classes e interfaces dependem (DepPack), as relações de produtividade são desproporcionais pelo fato de os termos de intercepto do modelo linear serem significativamente não-nulos para as métricas de: distância normalizada para a sequência principal (ND), relacionada ao princípio de abstrações estáveis (SAP); número de relações entre classes e interfaces (R), relacionada à complexidade do *software*; e de coesão relacional (H), relacionada à coesão do *software*.

Já a análise de regressão das relações de produtividade após a projeção das DMUs na fronteira eficiente evidenciou que o termo de intercepto foi significativamente não nulo para todas as relações analisadas. Desta forma, corrobora a desproporcionalidade das relações de produtividade aqui consideradas. Além disso, o modelo linear é ainda mais evidente do que para os dados originais das métricas de qualidade, com alta qualidade da previsão do modelo comparado com o seu nível de não precisão e alto poder de previsão em função do inverso do número total de operações normalizado (1/NNumOps_tc), para: o inverso do número de pacotes aos quais classes e interfaces dependem (1/DepPack); o inverso do número de relações entre classes e interfaces (1/R); e a coesão relacional (H). Este efeito possivelmente é produto do fato de o modelo DEA gerar uma aproximação da fronteira eficiente empírica por uma combinação linear por partes das observações mais eficientes. Porém, o modelo linear não se mostrou adequado para a previsão para o inverso da métrica de distância normalizada para a sequência principal (1/ND), provavelmente em decorrência da não-linearidade inerente à função módulo utilizada no seu cálculo.

Por fim, a análise estatística das medidas de eficiência da DEA através dos testes estatísticos de significância de Banker mostrou significativamente que: a nova versão da API (org.apache.hadoop.mapreduce) possui maior eficiência do que a versão antiga API (org.apache.hadoop.mapred), considerando todas as versões do *software* Hadoop; não houve variação na eficiência das APIs em decorrência da introdução da nova arquitetura (YARN) na versão 2.0.0 do *software* Hadoop, considerando ambas versões da API MapReduce; e que, com a introdução da nova arquitetura (YARN) na versão 2.0.0 do *software* Hadoop, houve uma inversão da eficiência em favor da nova versão da API, considerando cada versão da API MapReduce agrupadas por versão da arquitetura do *software* Hadoop.

6. Conclusões

Conclui-se, a partir da evidência empírica apresentada, que a análise por DEA não-relacional proposta por Fare e Lovell se mostrou capaz de avaliar a qualidade do projeto das APIs MapReduce do *software* Hadoop. Consideraram-se aqui os aspectos de complexidade, coesão, facilidade de reuso e aderência ao princípio SAP pela análise estática do código fonte, simultaneamente, como função do número de operações total da API como insumo. Além disso, conclui-se que tais avaliações podem ser utilizadas ao invés ou complementando a análise convencional por índices de produtividade individuais.

Usando meios apropriados de determinar mudanças estatisticamente significativas nas medidas de eficiência da DEA, dentre as versões do *software* Hadoop aqui consideradas e do ponto de vista estrito de qualidade das APIs MapReduce, verificou-se um favorecimento em direção ao desenvolvimento da nova API com a introdução da nova arquitetura (YARN), condizente com os esforços de descontinuação da API antiga. Além disso, de forma geral, a análise aqui apresentada foi capaz de evidenciar melhorias de qualidade no projeto do *software* decorrentes das funcionalidades adicionais providas pelo uso de classes abstratas e do padrão *context object* na nova API (org.apache.hadoop.mapreduce). Porém, em contextos onde utilizam-se versões do Hadoop dentre 1.0.4 e 0.22.0 é recomendável ainda utilizar a versão antiga da API (org.apache.hadoop.mapred).

Por fim, os modelos de regressão das relações de produtividade após a projeção das DMUs na fronteira eficiente representam potenciais modelos de referência para a previsão dos níveis eficientes para as métricas de qualidade do projeto das APIs MapReduce referentes ao nível de acoplamento dos seus pacotes (DepPack), nível de complexidade das relações entre os objetos dos seus pacotes (R) e nível de coesão relacional dos seus pacotes (H).

Sugere-se o desenvolvimento dos seguintes trabalhos futuros de forma a sobrepor as limitações identificadas neste artigo: aplicar a metodologia aqui apresentada em um estudo de casos múltiplos de forma a determinar qual dos dois padrões de projeto de APIs é preferível em um caso geral e se os modelos de regressão das métricas eficientes aqui obtidos ainda seriam válidos, considerando o ponto de vista da qualidade do *software*; utilizar técnicas meta-heurísticas de busca de forma a tentar deslocar a fronteira de eficiência empírica e sugerir possíveis melhorias para o projeto do *software* Hadoop, avaliando quais fatores seriam necessários para atingir um nível de qualidade das APIs MapReduce, superior ao previsto pelos modelos de regressão das métricas eficientes aqui obtidos, na nova arquitetura YARN.

Referências

Banker, R. D. e Kemerer, C. F. (1989), Scale Economies in New Software Development, *IEEE Transactions on Software Engineering*, 15, n. 10, 1199-1205.

Banker, R. D. e Natarajan, R., Statistical tests based on DEA efficiency scores, em Cooper, W. W., Lawrence, M. S. e Zhu, J. (Eds.), *Handbook on data envelopment analysis*. Springer US, New York, 273-295, 2011.

Charnes, A. e Cooper, W. W. (1985), Preface to Topics in Data Envelopment Analysis, *Annals of Operations Research*, 2, n. 1, 59-94.

Chatzoglou, P. D. e Soteriou, A. C. (1999), A DEA Framework to Assess the Efficiency of the Software Requirements Capture and Analysis Process, *Decision Sciences*, 30, n. 2, 503-531.

Gamma, E., Helm, R., Johnson, R. e Vlissides, J., *Design patterns: elements of reusable object-oriented software*, Addison-Wesley, Boston, 1994.

Martin, R. C., *Agile software development: principles, patterns, and practices*, Prentice Hall, New Jersey, 2003.

Paradi, J. C., Reese, D. N. e Rosen, D. (1997), Applications of DEA to measure the efficiency of software production at two large Canadian banks, *Annals of Operations Research*, 91-115.

Parthasarathy, S. e Anbazhagan, N. (2008), Evaluating ERP projects using DEA and regression analysis, *International Journal of Business Information Systems*, 3, n. 2, 140-157.

Schmidt, D., Stal, M., Hohnert, H. e Buschmann, F., *Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects*, John Wiley & Sons, Chichester, 2004.

Triantis, K. P., Engineering Applications of Data Envelopment Analysis, em Cooper, W. W., Lawrence, M. S. e Zhu, J. (Eds.), *Handbook on data envelopment analysis*, Springer US, New York, 401-441, 2004.

White, T., *Hadoop: The definitive guide*, O'Reilly Media, Sebastopol, 2012.

Zhu, J., *Quantitative models for performance evaluation and benchmarking: data envelopment analysis with spreadsheets and DEA excel solver*, Springer, New York, 2009.

http://archive.apache.org/dist/hadoop/core/, acessado em 14/02/2014

http://gephi.org/codeminer/index.html, acessado em 04/02/2013

http://www.sdmetrics.com/, acessado em 04/02/2013