# Applying Data Warehousing and Big Data Techniques to Analyze Internet Performance

T. M. S. Barbosa, R. Souza, S. M. S. Cruz, M. L. Campos and R. Les Cottrell.

SLAC National Accelerator Laboratory, Federal University of Rio de Janeiro, and Federal Rural University of Rio de Janeiro

Abstract- Measuring the quality of Internet is essential to evaluate the performance of data links around the world and to keep track of how countries have improved their connections throughout the years. Moreover, Internet performance measurements provide understanding for network bottlenecks, trouble-shooting and even insights about the impact of major events such as tsunamis, fiber cuts or social upheavals. For this reason, since 1998, the PingER (Ping End-to-end Reporting) initiative at SLAC National Accelerator Laboratory monitors end-to-end performance of Internet links spread over 160 countries, providing a worldwide history of Internet performance. Data containing network measurements are daily collected from PingER Measurement Agents (MAs) and stored into flat files. As a result, PingER maintains a valuable finegrained big dataset consisting of Internet performance data around the world. However, due to the large amounts of data, performing sophisticated joint analyses on those files may be so difficult that it becomes unfeasible in some scenarios. In this paper, we apply data warehousing techniques to transform the data on those flat files into structured data using a data model that facilitates complex analyses. We load the transformed data into a big distributed data warehouse that is able to perform complex analytical queries on large volumes of data in seconds. Finally, we show some data analyses correlating Internet performance data to hypothetical real-world scenarios.

Index Terms — Internet Measurement, PingER, Big Data, Data Warehousing.

#### I. INTRODUCTION

Measuring the quality of the Internet is essential to evaluate the performance of the data links around the world and to keep track of how countries have improved their connections throughout the years. Thus, it is essential to pursue novel computational techniques that provide informative analysis about the quality and the strategies being used by the countries in terms of networking. Even simple questions like "What is the overall rate of Internet connection speed in the world?" are hard to be answered because a large amount of data must be collected, treated and further analyzed.

T. M. S. Barbosa is with CAPES Foundation, Ministry of Education of Brazil, Brasilia, DF, 70.040-020, BR, on behalf of SLAC National Accelerator Laboratory, Menlo Park, CA, 94025, USA (phone: +1 650-549-6283; e-mail: tbarbosa@slac.stanford.edu)

R. F. Souza is with Federal University of Rio de Janeiro, Rio de Janeiro, RJ, 21941-901, BRAZIL (e-mail: renanfs@cos.ufrj.br)

S. M. S. Cruz is with the Federal Rural University of Rio de Janeiro, Seropedica, RJ, 23851-970, BR (e-mail: serra@ufrrj.br)

M. L. M. Campos is with Federal University of Rio de Janeiro, Rio de Janeiro, RJ, 21941-901, BRAZIL (e-mail: mluiza@ppgi.ufrj.br).

R. L. Cottrel is with SLAC National Accelerator Laboratory, Menlo Park, CA, 94025, USA (cottrell@slac.stanford.edu)

To cope with this, the IEPM group at the Stanford Linear Accelerator Center (SLAC) maintains the PingER (Ping Endto-end Reporting) project [18]. It monitors the end-to-end performance of Internet links worldwide. The project preserves a vast data repository of network performance measurements from and to sites all around the world. The repository contains data since 1998 and several associated applications have been developed and experimented in collaboration with universities and laboratories in South America, Europe, Pakistan and Malaysia.

Briefly, the project consists of gathering network data using the Ping facility and storing them into a centralized repository for future analyses. More specifically, at every 30 minutes, pings containing two different packet sizes are sent from over 80 MAs (source nodes) to over 700 locations (destination nodes) spread out over 160 countries. The data are associated with several network metrics (*e.g.*, throughput, packet loss, average round trip time) and stored into multiple semi-structured flat text files. The dataset now comprises over 100,000 files summing over than 60 Gigabytes. These network metrics can be analyzed to provide understanding for network bottlenecks, trouble-shooting, and even insights about the impact of major events such as tsunamis, fiber cuts or social upheavals.

Although storing historical data in flat files seemed to be a good solution at the beginning of the project, after seventeen years of hourly data gathering, the manipulation of such an amount of big data becomes a very challenging task when one needs to perform fine-grained data analyses to explore the entire dataset. Therefore, we need not only more efficient computational approaches for data management, but also more flexible structures for data analyses.

The above mentioned real-world scenario is quite close to the ones investigated by Data Warehousing technologies [14]. We advocate that similar solutions can be applied to the PingER project, providing a better understanding of analytical possibilities for its managers and users and providing a fair support for data aggregation and big data exploration.

As far as we are concerned, consolidated data management technologies that rely on centralized approaches are not suitable for computing the cumulative data volumes. For this reason, we propose a two-step approach that requires the utilization of distributed systems. Each step uses an opensource Big Data system. In the first step, we used the SciCumulus [9], a parallel Workflow Management System, to execute a Map-Reduce workflow to extract PingER legacy

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-76SF00515. data and to transform them following a multidimensional data structure. Secondly, we loaded the transformed data into an Impala analytical database [5], a big Data Warehouse (DW) system that runs on top of the Hadoop Distributed File System (HDFS) [1] offering flexible and scalable support to multi-user analytical *ad hoc* data queries. After loading data into Impala, we executed analytical queries, built data visualizations, and enabled complex analyses of PingER Internet performance data in an efficient and fast manner.

Regarding previous related work, Souza et al. [16] proposed a solution to enable PingER data to be publicly accessible using Semantic Web and Linked Open Data strategies. In order to do this, the authors similarly proposed a data transformation process to give structure and semantics to PingER data, linking these to external data, and also to load them into a Semantic Web data repository for further analyses. However, existing open source Semantic Web repositories cannot easily deal with large amounts of finegrained data. Hence, although they were able to provide public standardized and structured access to PingER data, only large granularity could be considered. Conversely, to face big data issues, we base this work on state-of-the-art open source big data technologies to consider PingER data in their finest granularity and to enable more detailed and sophisticated data analyses.

The remainder of this paper is structured as follows. Section II introduces the domain analysis of the real scenario studied and presents the theoretical background. Section III presents the proposed approach. Section IV shows the experiments and the analytical queries, highlighting the importance of managing and exploring these data. Finally, Section V presents the conclusion, current limitations of this work and future work.

#### II. BACKGROUND

# A. PingER Project

The PingER project [18] was initiated in 1995 and has publicly accessible data back to the beginning of 1998. It uses the ubiquitous Ping facility to make Round Trip Time (RTT) and loss measurements from over 60 MAs in 23 countries to over 700 destinations in over 160 countries comprehending over 99% of the world's Internet-connected population. The measurements from each MA (or source) are scheduled at approximately 30 minutes intervals. The data from the MAs are daily uploaded to a centralized repository of text archives, and further analyzed to extract 16 different metrics [17], which include: the minimum, the average, and the maximum RTTs; jitter; unreachability (a target is defined as unreachable if it does not respond to any pings); the throughput; and the Voice over IP quality. The collected data are also aggregated into various time bins including hourly, daily, monthly and yearly data files.

A typical PingER Internet performance data measurement is defined by a combination of several network parameters like source node, destination node, timestamp, size of the ping packet (currently, PingER uses packet sizes of both 100 and 1000 bytes), and the target metric. The values of measurements are stored in flat text files. The filenames follow a specific naming pattern that combines the date of the measurement, packet size and name of evaluated metric. For instance, the daily data files contain hourly measurement data (average value), one record for each combination of source and destination nodes. In addition to daily data files, there are smaller and fine-grained files with the average values for all days in a year.

```
<metric>-<size>-by-<site|node>(-<YYYY>?)
(<mm>?)(-<dd>?).txt.gz
<metric>-<size>-by-<site|node>
<60|120|365>days.txt.gz
<metric>-<size>-by-<site|node>
<allmonths|allyears>.txt.gz
```

E.g. throughput-100-by-node-2005-05-01.txt.gz

PingER harvested data are retrieved using a Web application named Pingtable, which provides a parameterized GUI capable of loading the raw data from the flat text files (Fig. 1). Despite being a simple and straightforward strategy for creating the datasets, it presents limitations for data maintenance and cannot cope with sophisticated analytical processing operations, such as data aggregations, because the data are spread over many unconnected files.

The original PingER project prepares the output for the most expected queries accessing the original data or even preprocessed flat files. Thus, one you want a given result, he have to write a script to get it, extracting the required data and providing its presentation. Such approach not only does this require a detailed knowledge of the data, knowing where to find and access the relevant data, but also knowledge how to write the scripts. This process can take days or weeks. Thus, these are key motivations to investigate and propose the development of a novel distributed approach that should provide innovative answers in terms of reduced processing time, enhanced query support and novel user functionalities.

### B. Foundations of Data Warehousing

To populate a DW, a dimensional data model must be conceived *a priori*. After that, we need to implement a dataflow process called Extraction, Transformation and Loading (ETL), very well-known in the data management research field. In this process, data is selected, extracted, transformed following the dimensional data model, and stored into a DW for enhanced data analyses. Moreover, another important aspect needs to be highlighted.

lake selections on current datase	et:													
	· By by-node · For Pa		ytes) 100	· Tick-										
rom EDU SLAC STANFORD PING	ER To WOR	RLD			• Es	cluding N	one							
how all • nodes. Change the	e dataset hep * Only sh	low sites wi	th any *	datapoint	s. Show dat	a for disabi	led nodes too	- El .						
lter Data 🗏 (monthly, alimonthly	& allyearly data only. U	ses afs/slac	g www.'n	ww-iepm	pinger/slac	only/pingta	ble_filter cfg	) Load in	nto table					
Ping Response Times			Click on					Please Note					-	
less than 62.5ms are shown black. 52.5ms to 125ms are shown grown.		Each C	Each Column Header to sort by that column				These	hues show	n here are	for the ful	I day GM	T tunte	_	
			Each Column Header to sort by that column Monitoring Site to graph the data with (only for SLAC A B in the ? column indicates this a Beacon host (monitor							red .				
25ms to 250ms are shown or ange			monitoring Site to graph the data with (only to SLAC A b in the Column indicates that a beacon nost (monitore) monitoring host & By=by-node) Smokering by most Monitoring hosts).											
250ms to 250ms are shown or ange.			The ? on each line to get access to information on the two hosts. An M in the ? column indicates it is a monitoring host.											
100ms or more are shown red.			plus access to the latest data, recent time series and frequency. An r in the 7 column indicates it is a monitoring new							ind.				
												or (anomico		
					utor access			1.000						
		traceros				recent o								
other Help is provided about the	tables and there is a Testo	rial Aleas	es the Dim	ER Home	mane Thus	report to a	he provided	n downle	and able tal		d. colors ( t	(a) format	for man	ist.
icel. See here for SEECS Pingtab	le pl and Latest Offsite C	onnectivity	Status see	n from SL	AC.			-		Cangentine				
Motion Bubble Chart Matri	x of Monitoring vs Mon	itored Regi	ions Pine	er number	of nodes/c	ountry	Google	map of si	ites	Visits	to this pa	100		
Line and Lin														
				5	1		88.Y.	945	00	- 8	-			
			-							Clur	str Vaps*	2 2 2		
torine-Site Rente	-Site	1	2012015	Jun 2015	Rev2015	Apr:2015	Par2015	516 ( Feb2815	Jan2015	Chur	str Vaps	RC12014	Sep2014	App
J.SLAC.STAMFORD.PINGER AD.UDA			2u12015 176.775	2un2015 176.022	Kay2015. 175-320	Apr:2015 177:145	Rac2015.	5 cb 2015	Jan2015	Dec2014	St/Vaps St/Vaps Sov2014 179.781	RC12014	Sep2014 176.624	
.SLAC.STANFORD.PINGER AD.UDA	NI OLICE.GOV.NI	20		176.822	May 2015. 175-329	177.145	177.828	179:510	176.363	Dec2014 120.775 201.225	Str Vaps Str Vaps Sov 2014 179.781 pr2.545	0ct2014 176.884 277.019	Sep2014 176.624 388.43	
I.SLAC.STANFORD.PINGER AD.UDA I.SLAC.STANFORD.PINGER A8.ADP I.SLAC.STANFORD.PINGER AE.ZU.	4.N1 IOLICE.60V.N1 AC.N2	10 20	176.775	176.422	Mag 2015. 175.329- 285.519	Apr:2015 177-145 177-1956	177.828	179:510	176.363	Dec2014 128.775 241.229 774.189	Str Vaps Str Vaps Str 2014 177.545 272.598	Rct2014	Sep2014 176.624 388.417 308.555	
I.SLAC.STANFORD.PINGER AD.UDA I.SLAC.STANFORD.PINGER AB.ADP I.SLAC.STANFORD.PINGER AE.20. I.SLAC.STANFORD.PINGER AF.MUL	A.NI HOLICE.GOV.NI AC.N2 TINET.NI	11 11 11		176.822	Rey2015. 19. 19. 19. 19. 19. 19. 19. 19. 19. 19	177.145	177.828 273.948 559.904	179:510 185:417 669:930	176.363	Dec2014 220-775 220-725 220-220 274-510 321-410	80x2014 1275.781 1272.545 272.698 113.629	0ct2014 176.604 277.019 277.142 115.819	Sep2014 176.624 288.417 388.556 331.406	
J.SLAC.STANFORD.PINGER AD.UDA J.SLAC.STANFORD.PINGER AF.ADP J.SLAC.STANFORD.PINGER AF.MUL J.SLAC.STANFORD.PINGER AF.MUL J.SLAC.STANFORD.PINGER AF.NDL	A.NI POLICE.GOV.NI AC.N2 TINET.NI MA.NI	21 21 21 28	117.457 117.457	176.822 202.164 325.464 129.695	Rey2015 176-320 1 205-929 206-441 140,463	177.145	177.828	179:510	176.363	Dec2014 126-775 246-229 724-510 125-439 355 055	80x2014 378,781 379,781 372,545 312,545 313,654	9ct2014 176.000 277.000 277.142 119.019 119.756	Sep2014 176.624 380.432 331.496 131.496 195.174	
I.SLAC.STANFORD.PINGER AD.UDA I.SLAC.STANFORD.PINGER AE.ADP I.SLAC.STANFORD.PINGER AE.ZU. I.SLAC.STANFORD.PINGER AF.MUL I.SLAC.STANFORD.PINGER AF.NDU I.SLAC.STANFORD.PINGER AF.PPU	4.N1 OLICE.GOV.N1 AC.N2 TINET.N1 HA.N1 F.GOV.1880	11 11 11 11 11 11 11 11 11 11 11 11 11	176.775	176.422	Ray2015 175-229 255-579 255-445 190-453	177.145	177.828 177.946 559.904 141.110	179:530 265:437 669:930 517:555	176.363	Dec2014 120.775 201.220 174.100 133.430 133.430	Str Waps*	9ct2014 176.804 277.009 277.009 277.519 119.716	Sep2014 176.624 388.412 388.956 195.124 226.358	
J. SLAC, STANFORD, PINGER AD. UDA . SLAC, STANFORD, PINGER AD. ADD . SLAC, STANFORD, PINGER AC. PUL . SLAC, STANFORD, PINGER AC. MUL . SLAC, STANFORD, PINGER AF. MUL . SLAC, STANFORD, PINGER AF. MOD . SLAC, STANFORD, PINGER AC. ALL	4.N1 OLICE.GOV.N1 AC.N2 TINET.N1 IA.N1 .GOV.MAN MANIAONLINE.NET.N1	11 11 11 11 11 11 11 11 11 11 11 11 11	117.457 117.457	176.822 202.164 325.464 129.695	Ray2015 175.329 285.579 336.445 190.433 224.099	177.145	177.828 273.948 559.904	179:510 185:417 669:930	176.363	Pec2014 235-775 241 - 221 774 - 510 125 - 405 125 - 405	str Vaps 577.781 577.781 577.545 517.545 517.545 217.505	0ct2014 176.604 277.010 272.142 119.010 119.756	Sep2014 176.624 388.417 388.596 331.404 195.174 - 226.258	
J.S.LAC.STANFORD. PTINETS AD. UND J.S.LAC.STANFORD. PTINETS AD. AD. J.S.LAC.STANFORD. PTINETS AD. AD. AD. J.S.LAC.STANFORD. PTINETS AD. AD. AD.	4.N1 OLICE.GOV.N1 AC.N2 TINET.N1 HA.N1 F.GOV.1880	11 11 11 11 11 11 11 11 11 11 11 11 11	117.457 117.457	176.822 202.164 325.464 129.695	Ray 2015 175-329 285-579 336-445 348-45 348-	177.145	177.828 177.946 559.904 141.110	179:530 265:437 669:930 517:555	176.363	Dec 2014 126-775 261-229 251-439 131-439 131-933 217-733 249-181	80x2014 378.781 377.781 377.545 372.545 313.834 217.505 244.960	0ct2014 176.804 277.804 277.142 135.819 139.758 248.969 107.829 247.306	Sep2014 176.624 381.551 331.406 195.774 226.258 201.155 343.076	175 217 207 207 207 207 207 207 207
J.S.LAC, STANFORD, PTINES         AD. ION           J.S.LAC, STANFORD, PTINES         AI. AND           J.S.LAC, STANFORD, PTINES         AF. AND           J.S.LAC, STANFORD, PTINES         AF. AND           J.S.LAC, STANFORD, PTINES         AF. MO	. M1 OLICE.60V.N1 AC.N2 TINET.N1 JAL.N1 JALTAON.INE.NET.N1 JANTAON.INE.NET.N1 JANTPASSY.VMM A.N1		117.457 117.457	176.822 2172.164 535.404 139.075 548.487 202.657 227.289 224.510	216.657	177-145 277-006 104-255 556-239 214-568 217-595 219-150	177.828 272.944 559.904 141.110 211.968 254.962 224.731	179,510 105,417 669,930 117,555 216,999 252,571 235,783	176.363 188.839 218.815 548.815 348.815 229.648	Pec2014 120,775 201,220 121,430 133,635 217,733 249,101	8072014 379-781 379-781 377-545 113-620 113-620 113-620 113-620 113-620	0ct2814 176.894 277.5894 277.542 319.819 199.756 218.969 197.829 247.366	Sep2014 176.624 388.437 388.594 311.406 105.124 286.258 201.105 243.076	175 217 207 207 207 207 207 207 207
5. SLAC, STANFORD, PTIMER AD, UM SLAC, STANFORD, PTIMER AF, AD SLAC, STANFORD, PTIMER AF, AD	. M1 90.TCE.60V.N1 AC.M2 TTNET.M1 N.M1 NGV.NBA NAM2AGN.TME.NET.M1 NAM2.M3 TAMEPHBASSY.NBA N.M1 U.MA	11 11 11 11 11 11 11 11 11 11 11 11 11	176.775 117.407 126.377 540.325 208.397 215.009 218.999 214.942	176.822 2011.564 525.654 529.655 540.657 202.657 227.285 224.510 227.127	216.657 217.481	177-145 237-096 104-255 356-299 214-568 217-591 219-150 216-845	177.828 272.944 559.964 141.119 211.968 234.962 224.731 228.968	179,510 105,417 669,930 117,555 216,999 252,571 235,783 245,242	176.363 188.839 218.815 548.815 348.815 229.648	Des2014 120:775 231:2715 121:405 123:405 123:405 121:733 249:101	Ser Waps Ser Waps Ser Vaps 1272-545 2772-545 112-670 1133-855 241-560 241-560	9ct2014 176-804 277,000 277,000 277,142 119,275 119,275 218,969 247,396	Sep2014 176624 136412 386412 386556 313406 313406 313406 313406 313406 326258 243076	179 210 200 335 200 200 200
1, 1, 2, 4, 2, 7, 144 FORD, P, 1106E         AD, 100           1, 2, 4, C, 2, 144 FORD, P, 1106E         AL, 200           1, 2, 4, C, 2, 134 FORD, P, 1106E         AL, 200           1, 2, 4, C, 2, 134 FORD, P, 1106E         AL, 200           1, 2, 4, C, 2, 134 FORD, P, 1106E         AL, 200           1, 2, 4, C, 2, 134 FORD, P, 1106E         AL, 200           1, 2, 4, C, 2, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 2, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400           1, 2, 4, C, 3, 134 FORD, P, 1106E         AL, 400	. MI AC. N2 AC. N2 TTNET. N1 A. N1 A. OT ANTION. THE NET. N1 ANTION. THE NET. N1 TANEPBASSY. VBM M. N1 U. MAJ BLET. N1		176.775 117.407 126.577 540.325 208.397 215.009 218.009 218.009 218.009 218.042 129.572	176.822 2011.164 525.604 525.604 129.605 560.627 202.657 227.289 224.510 227.127 129.208	216.657 217.481 128.958	177:145 237:000 104:255 359:209 214:568 217:593 219:150 216:845 129:941	177.828 272.946 559.904 344.110 211.968 234.962 224.985 130.092	179:518 205:417 669:938 517:555 216:999 232:571 235:783 245:282 129:516	176.363 288.899 210.315 548.815 229.648 244.871 128.897	Precional 2014 2014 2014 2014 2014 2014 2014 2014	500 Vaps 500 Va	9212014 176.804 176.804 277.000 177.000 197.809 197.809 197.809 197.809	Sep2014 176.624 188.412 381.594 131.406 195.174 226.258 201.195 245.076	175 217 207 207 207 207 207 207 207 207 207
5. 5.4.6. STANPORD, PTHERE 40, UAD 5. 5.4.6. STANPORD, PTHERE 44, UAD	. MI AC. M2 AC. M2 TTHET. NI M. MI J. GOV. MAR J. GOV. MAR J. AND MI TANTPERSY MAR M. NI J. MAR M. MI M. MI M. MA MI M. MI M.	<u>此此推在随</u> 此推出社社社社	176.775 117.407 326.577 540.325 208.397 215.009 218.099 218.099 218.099 218.099 218.099 218.099 218.072 209.753	176.827 2011.164 336.404 129.605 340.407 202.657 227.289 224.510 227.127 129.206 341.934	216.657 217.481 128.958 125.296	177.145 277.006 364.255 356.739 214.568 217.593 216.855 129.541 327.499	177.828 273.944 559.904 943.110 211.968 234.962 224.231 224.231 228.366 136.692 137.819	179:538 205:437 669:938 517:555 216:999 232:571 235:783 245:242 129:516 524:785	176.363 188.839 218.815 548.815 348.815 229.648	Dec2014 238-775 274-229 274-199 333-439 333-439 277-733 249-101 138-545 125-545	str Waps 578,781 579,781 577,545 577,545 577,545 513,556 244,960 131,2795 131,2795 131,2795	0c12014 1776.804 277.010 177.809 177.8	Sep2014 176.624 388.417 388.556 391.556 201.258 201.155 243.258 138.651 128.651 124.130	175 217 207 207 207 207 207 207 207 207 207
0.1.544.CTMMP700.PTME84 40.149 0.1.544.CTMMP700.PTME84 42.74 0.1.544.CTMMP700.PTME84 42.74 0.1.544.CTMMP700.PTME84 42.74 0.1.544.CTMMP700.PTME84 42.74 0.1.544.CTMMP700.PTME84 42.74 0.1.544.CTMMP700.PTME84 44.44 0.1.544.CTMMP700.PTME84 44.44 0.1.544.CTMMP700.PTME84 44.44 0.1.544.CTMMP700.PTME84 44.44 0.1.544.CTMMP700.PTME84 44.74 0.1.544.CTMMP700.PTME84 44.74	. M1 AC. M2 AC. M2 TTNET. N1 AC. M2 TTNET. N1 A. M1 AC. M2 AC. M2 TTNET. N1 AC. M2 AC. M2	近 近 近 近 近 近 近 近 近 近 近 近 近 近 近 近 近 近 近	126.775 117.407 136.577 540.135 208.397 215.009 218.009 218.099 218.099 218.999 218.995 219.757 196.757	176.822 2011.164 336.005 346.005 346.012 202.657 227.289 224.510 227.127 129.208 341.954 350.018	216.657 217.481 128.958 125.296 206.095	177.145 277.006 104.255 356.239 214.568 217.591 210.150 216.845 129.941 327.499 205.639	177.828 273.946 559.944 141.119 211.968 234.962 224.731 228.386 130.092 377.819 277.217	179:538 205:437 669:938 817:555 216:999 232:571 235:783 245:242 129:516 524:785 276:856	176.363 288.899 210.315 548.815 229.648 244.871 128.897	Pes2014 128-775 281-229 724-138 323-439 33-439 33-439 33-439 33-439 33-439 33-439 33-439 33-439 33-439 33-439 33-439 33-439 33-439 249-149	Str Vaps Str Vaps Str Vaps Str 2014 Str 20	0ct2014 176.804 277.04 135.809 277.142 135.809 218.060 218.060 247.306 132.060 247.306	Sep2014 176.624 388.47 381.594 331.406 105.174 226.258 243.076 158.651 158.651 154.130 235.956	175 217 207 207 207 207 207 207 207 207 207
0. USAC STREPTON PTHEME 40. UND USAC STREPTON PTHEME 41. AND USAC STREPTON PTHEME 41. AND USAC STREPTON PTHEME 42. AND USAC STREPTON PTHEME 42. AND USAC STREPTON PTHEME 42. AND USAC STREPTON PTHEME 44. AND USA	1.111 OLICE, 600, 141 AC, 102 ITTHET 143 IA, 411 1.600, 1404 IA, 414 IA, 407 IA, 407	<u>此此</u> 進出進出於 出進 北 進 出 進 北 進 上 進 上 進 上 進 上 進 上 進 上 進 上 進 上 進 上 進 上 進 上 進 上 進 上 二 一 一 一 一 一 一 一 一 二 一 一 一 二 一 一 一 二 一 一 一 二 一 一 一 一 一 一 一 一 一 一 一 一 一	1176.775 117.467 126.577 540-525 208.397 215.009 218.009 234.942 129.572 209.753 356.753 313.157	176.822 2172.564 535.604 535.604 529.605 540.607 202.657 227.289 224.510 227.127 129.200 511.934 523.657	216.657 217.481 128.958 125.298 206.095 189.542	177.145 277.006 104.055 359.739 214.568 217.593 219.159 216.845 129.941 327.499 216.839 717.635	177.828 272.944 559.904 343.110 211.968 234.962 224.731 228.386 130.092 327.819 277.217 333.511	179:510 265:437 669:930 147:455 216:999 232:571 235:783 245:282 129:516 224:783 276:85 204:543	176.363 268.809 318.315 548.818 229.648 244.821 128.897 528.215 291.892 105.289	Pec2014 326.775 326.729 724.5181 323.430 235.430 249.101 327.733 249.101 138.345 125.945 125.945 239.145 239.145	Str Waps Str Waps Str Vaps Str 2014 Str 781 Str 782 Str 782	0ct2014 176.004 277.004 277.004 199.798 197.829 247.006 197.829 197.429 197.421 197.421 197.421 197.421	Sep2014 176.624 388.413 388.556 331.406 105.174 206.258 203.105 243.076 158.651 124.130 203.556 128.4130 203.556 128.4130	175 200 200 200 200 200 200 200 200 200 20
0. USAC STRUMENCE PTHEES 40. USA USAC STRUMENCE PTHES 41. 400 USAC STRUMENCE PTHES 40. 400 USAC STRUMEN			1276,775 117,407 135,577 340,325 208,397 215,009 214,942 129,372 209,733 356,735 356,537 255,402	176.822 2111.164 325.604 129.605 366.62 202.657 227.289 224.510 227.127 129.206 501.934 255.605 311.934 255.651	216.657 257.485 128.958 125.228 286.095 387.582 263.512	177.145 277.036 104.255 356.739 214.568 217.595 216.865 129.645 527.409 705.829 705.829	177.428 272.944 559.904 944.110 211.968 234.962 224.231 228.366 130.492 137.419 277.217 333.521 229.466	179:510 185: 437 669,930 847,455 216,999 232:571 235:783 245:282 129:516 129:516 129:516 129:516 129:517 129:517 129:575 129:5777 129:577 129:577 129:577 129:577 129:577 1	176.363 288.899 210.315 548.815 229.648 244.871 128.897	Pec2014 138, 275 281, 218 221, 418 221, 418221, 418 221,	Str Waps Str Waps 327.781 327.781 327.781 327.595 313.654 237.595 131.295 135.295 1	0ct2014 176.004 177.009 277.147 119.758 247.809 247.809 247.809 247.809 132.009 247.506 132.009 247.506 132.009 247.506	Sep2014 176.624 188.595 191.596 195.174 226.258 243.076 158.651 154.130 291.595 245.976	175 200 200 200 200 200 200 200 200 200 20
0. USA USA TEMPERATURA TAL AND USA		ununununununununununununununununununun	176.775 117.407 126.577 540.525 208.397 215.009 214.905 215.009 214.909 214.905 215.009 214.905 215.009 214.905 215.009 214.905 215.009 214.905 215.009 214.905 215.009 214.905 215.009 214.905 215.009 214.905 215.009 214.905 215.009 215.009 214.905 215.009 214.905 215.009 214.905 215.009 215.009 214.905 215.009 215.009 215.009 215.009 215.005 215	176.822 202.164 355.494 129.695 360.672 202.657 227.289 224.510 229.207 129.208 523.872 203.619 129.208 523.872 270.508	216.657 257.485 128.958 125.236 266.095 365,557 263.955 267.788	177:145 237:006 104:255 350:739 214:568 219:159 216:858 219:159 216:859 216:859 217:699 216:839 217:699 248:457	177.828 272.944 559.904 944.110 211.968 234.962 224.251 224.355 139.092 377.217 335.511 229.465 232.467	129:510 105:417 669.930 117:555 216:999 232:571 235:783 245:282 129:516 129:516 129:516 129:517 129:510 129:510 120:715 120	176.363 240.839 310.315 340.816 229.640 244.871 128.897 370.715 291.807 346.946	Pec2014 128.7775 238.2201 234.220 234.230 133.033 125.430 133.033 125.430 125.9100 125.91000 125.91000 125.91000000000000000000000000000000000000	507 Waps* 507 Waps* 507 Waps* 507 2014 507 2014 507 2014 517 2015 518 200 513 2005 518 200 519 2005 518 2	0ct20314 176.809 277.0000 277.0000 277.0000 277.0000 277.0000000000	Sep2014 176-624 188.e17 286.556 195.124 226.258 201.155 245.076 138.651 124.130 295.254 218.651 124.130 295.254 218.254 218.254 218.254	2000 2017 2017 2017 2010 2010 2010 2010
M. SLAC, TATMINGO, P.THESIB         M. UNICAL           M. SLAC, STANDORD, P.THESIB         M. ET           M. SLAC, STANDORD, P.THESIB			1276,775 117,407 135,577 340,325 208,397 215,009 214,942 129,372 209,733 356,735 355,402	176.822 2111.164 325.604 129.605 546.62 202.657 227.289 224.510 227.127 129.206 501.934 503.652 203.655	216.657 257.485 128.958 125.228 286.095 387.582 263.512	177.145 277.036 104.255 356.739 214.568 217.595 216.865 129.645 527.409 705.829 705.829	177.828 272.946 559.944 945.119 211.968 234.962 224.231 228.388 3.38.962 322.838 3.38.962 322.838 3.38.962 333.511 229.465 232.460	179:510 185: 437 669,930 847,455 216,999 232:571 235:783 245:282 129:516 129:516 129:516 129:516 129:517 129:517 129:575 129:5777 129:577 129:577 129:577 129:577 129:577 1	176.363 268.809 318.315 548.818 229.648 244.821 128.897 528.215 291.892 105.209	Pecca014 238-775 248-229 274-218 222-430 233-825 274-735 224-521 239-545 239-1410 249-1410 249-1410 249-545 219-1410 249-5455	131.295 138.402 275.755 348.766 222.946 233.844 257.855	0c12014 1776.004 277.000 277.142 315.819 139.72 248.069 247.142 247.14	Sep2014 176-624 188, 412 235, 234 235, 236 236, 258 245, 258 245, 258 245, 258 245, 258 245, 258 234, 139 253, 254 254, 259 254, 951 256, 256 256, 951 256, 256 256, 951 256, 256 256, 951 256,	175 200 200 200 200 200 200 200 200 200 20

Fig. 1. Screenshot of Pingtable, the web application used to retrieve data from the PingER repository.

Due to the complexity of the scenario, the amount of data involved and the number of data transformations that happen in a complex dataflow, keeping track of how each specific item was transformed is important for reproducibility, failure analyses, missing data discovery, or anomaly detection. For this reason, using a system that not only transforms big data using distributed approaches is important, but also keeping provenance metadata (i.e. the history of data transformations dataflow) in a complex can bring significant advantages[13][8].

#### C. Technology choice

This work shares characteristics of analytical environments and deals with huge amounts of data, in particular, PingER Internet performance measurement data. To avoid drawbacks like the use of proprietary distributed data warehouse solutions, which are commonly highly priced or require specific parallel hardware architectures, we set requirements to select the software tools used in this work. The requirements are that the data warehouse system should be able: (i) to execute in a commodity cluster, rather than a

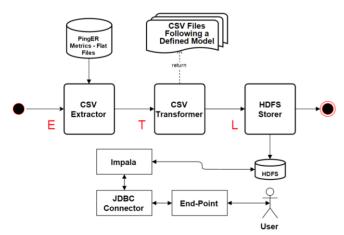


Fig. 2. Conceptual architecture of PingER dataflow.

specific parallel hardware; (ii) to deal with analytical processing; (iii) to deal with huge amounts of data; (iv) to be scalable; and (v) to be a low-cost solution based on stable open-source software.

The technical choice was SQL-on-Hadoop approaches. Thus, we evaluated four alternatives: (i) Apache Hive – Data are stored in HDFS, Hadoop MapReduce is used to execute analytical queries, and an SQL-like language is available[2]; (ii) HadoopDB - It consists of a common database management system (DBMS) in each node, supporting the cluster of databases, and it uses Hadoop MapReduce to manage and execute the queries [10]; (iii) Apache Tajo – It shows better performance than Hive, as it has a distributed query optimization approach; it does not use Hadoop MapReduce for query execution, and it supports standard SQL [3]; (iv) ElasticSearch - it is a two-way connector with Hadoop that allows users to make real-time searches, capable of executing queries and perform big-data analytics. ElasticSearch also makes use of Hadoop MapReduce to perform its queries [11]; and (v) Cloudera Impala – similarly to Tajo, it presents better performance than Hive, it does not use MapReduce, and it has a query optimization approach; however, Impala is older with improved funcionalities, providing a much more extensive documentation and development facilities [5].

The Hadoop MapReduce jobs are, by default, nonoptimized and scan oriented, which impairs many common DBMS tasks, especially when dealing with heavy-weight queries, common in an analytical scenario [12]. Based on that, we decided to use approaches that do not rely on Hadoop MapReduce to execute queries. Thus, only Cloudera Impala and Apache Tajo were adequate candidates. However, Cloudera Impala was chosen because of its maturity, ease of access and use, and the existence of proper documentation.

For the distributed extraction, transformation and loading (ETL) process, approaches that rely on Hadoop MapReduce could be used. However, in this case, it would be hard to relate the resulting transformed data to the original raw data since they do not store provenance metadata. Moreover, none of them store execution data at runtime in a data repository that enables runtime execution analyses and monitoring, which can facilitate long executions, common in big data analytical scenarios [8]. For this reason, we have adopted SciCumulus [9], which facilitates parallel workflow executions on distributed environments and stores retrospective provenance metadata at runtime. Although we could have used another approach, such as defining the data schema using ElasticSearch, we choose to make use of the workflows, because Workflow Management Systems are able to keep tracking of the data and also collect provenance about the data transformations.

# III. PROPOSED APPROACH

In Section II.A, we introduced the PingER project, explaining which Internet performance data are collected, and

how these data are stored. We have also argued that storing a large amount of data in multiple files is not efficient for big data analyses. In this section, we depict our proposed solution that facilitates fine-grained data exploration, presenting the designed dimensional data model and the ETL MapReduce dataflow, highlighting the loading process into Impala.

#### A. Data Selection and Dimensional Data Modeling

Our approach starts by selecting which data will be treated in the ETL process. Regarding data granularity, we explained (Section II.A) that PingER stores hourly data in daily files and also stores aggregated results in smaller files. In this work, we selected the finest granularity to process, *i.e.*, the daily files that contain hourly collected data. These finestgrained files contain the biggest data volume. The data was gathered from 1998 to 2014 (resulting in more than 60 Gigabytes of data files). The text files containing aggregated (such as monthly and yearly) results contain less than 10 Megabytes and do not need to be processed because we can run trivial SQL queries to derive the aggregated results from the finest-grained data.

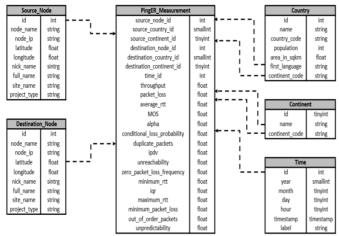


Fig. 3. The PingER dimensional data model used in the data warehouse

To define a dimensional data model, we need to identify network measurements and their perspectives of analysis. As previously explained, a ping measurement is defined by a combination of parameters (source, destination, timestamp) and associated with specific network metrics. This combination defines one measurement occurrence or, as it is called in DW theory, a fact. Additionally, each parameter corresponds to a DW dimension in a dimensional data model. Besides, as each source or destination node is physically located in a country and continent, we also define those as DW dimensions, to enable aggregations by regions (countries or continents) directly relating them to the PingER measurement fact table. The proposed dimensional data model is illustrated in Fig 3.

# B. Process for Data Transformation and Loading into the Data Warehouse

After designing the dimensional data model, we need to define the data transformation process. For this, we modeled

a MapReduce dataflow to be executed in SciCumulus. The dataflow is composed of two activities: a mapper and a reducer. The mapper reads the selected raw PingER text files and transforms them following the designed dimensional data model. More specifically, each mapper invocation reads a text file containing measurement data for a given metric and day of the year, and transforms this file into a resulting file that contains the same information, but follows a dimensional structure.

The reducer simply combines all files for a given year into a single yearly big file. Each yearly file size corresponds to the sum of all file sizes for a particular year. Using the Workflow Management Systems, all distributed process was transparent and taken care of by the workflow system. Moreover, we could monitor the status of the execution and how each specific data transformation occurred, *i.e.*, which and how each specific raw data file was processed to generate a structured file for a given combination of parameters. This facilitates the transformation process in a long data transformation run. Finally, those yearly files could be easily inserted into Impala, the distributed data warehouse for big data analyses.

# IV. EXPERIMENTAL EVALUATION

To run the experiments, we used a cluster consisting of four virtual machines with four cores, 16 GB of RAM and 220 GB of Hard Drive storage each. In total, there were 16 cores, 64 GB of RAM and 880 GB of storage with a gigabit/sec LAN connection. The operating system used was Linux RedHat 6.6 [15]. We used the Cloudera Distribution of Hadoop (CDH) 5.4.4 [4] and the SciCumulus [9]. We first ran the data transformation process, then loaded the resulting files into Impala, and finally executed multiple analytical queries to analyze the data.

For the transformation process, we ran the dataflow depicted in Section III.B. The mapper activity transformed over 100,000 flat files into the same number of comma separated value (CSV) files, but each of which followed the designed dimensional data model. The reducer activity combined those transformed files into 17 large text files, summing 45 Gigabytes of transformed files. The entire transformation process took 6h 12 min to run.

After that, those 17 files were loaded into HDFS and then inserted into Impala. The data directories were created in the HDFS using the Hadoop commands via Linux command line. The fact table was partitioned in 16 files, one file for each year. One directory was created for the DW dimensions (one subdirectory for each dimension) and one directory was created for the facts (one subdirectory for each partition). After that, the generated data was also loaded into the file system using the following commands:

```
hdfs dfs -mkdir /pinger/csv/dimensions/
```

```
hdfs dfs -mkdir
/pinger/csv/dimensions/country
```

```
hdfs dfs -put country.csv
/pinger/csv/dimensions/country
```

Each file took 32 seconds on average to be uploaded to HDFS. Once there, we created an external table linking the CSV dimension file with Impala. An external table uses arbitrary HDFS directories, where the data files are typically shared between different Hadoop components [6]; it works as a link between Impala and HDFS. The external tables took only 0.28 seconds on average to be created on Impala using statements, such as:

After we had created the external tables, the Impala tables were created using Parquet format. Among the available formats (plain text [CSV], Parquet, Avro, RCFile or SequenceFile), Parquet is as a column-oriented binary file format that was created to be highly efficient for large-scale queries [7].

This fits exactly with what we need in our analytical environment. The creation of the tables took 0.28 seconds on average, and it took 6.75 seconds, on average, to have the data inserted. Note that all actions on an Impala table reflect on the data in the HDFS.

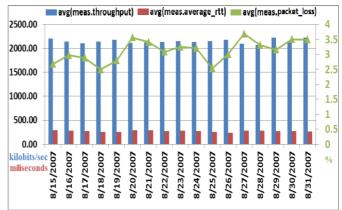
```
create table time like time_csv
    stored as parquet;
insert into time select * from time csv;
```

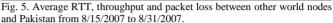
After uploading and inserting all data into the tables, we finally ran the queries over the data. As expected, the environment allowed us to run complex analytical queries in just a few seconds.

In Fig. 4, for instance, we could calculate the average throughput. This supports analysis about a very specific range of time, which would be very difficult in the original PingER. These results could also be correlated with another data to do some very specific analysis, as for instance analyzing the use of the Internet during certain months, or during specific holidays.

Also in Fig. 4, we show the analysis involving the average throughput and round trip time between 1999 and 2001, in a very specific time range (in this case, the months of January and December of each year).

Another interesting example of data investigation would analyze data from specific scenarios. For example, suppose a hypothetical scenario that considers that an earthquake has happened in Pakistan on July 23<sup>rd</sup>, 2007, and we want to analyze the performance of Internet in that country 8 days





before and 8 after days the quake. We could just run a standard SQL query in order to retrieve the data (throughput and RTT) that we need.

SELECT	t.day,	avg	(meas.t	hroughpu	t),
avg(meas	.average_rtt),	avg(m	eas.pac	ket_loss	)
FROM	pinger_measure	ment	meas,	time	t,
country	dst_country				
WHERE	<pre>meas.time_id =</pre>	t.id			
AND	t.time_stamp	BETW	EEN	2007-08	-15
15:00:00	' AND '2007-08-	-31 18	:00:00'		
AND ds	t_country.count	ry_co	de = 'PI	κ'	
AND me	as.year = 2007				
GROUP	BY t.day, dst_c	country	y.id		
ORDER	BY t.day, dst_c	country	y.id;		

This is a standard SQL query created to analyze the cited scenario. In Fig. 5, we show the analysis containing the throughput and the average RTT of each day, individually. Other metrics could be used, such as the packet loss or the reachability. The proposed solution performs very well and allowed us to run queries using the entire data from PingER in just few seconds.

In Table 1, we can see the average throughput and average

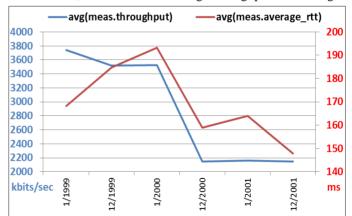


Fig. 4. Average throughput and RTT between US and the rest of the World from 1999 to 2001 on January and December.

packet loss retrieved from data of all PingER monitors over the world. Retrieving all those data just took 19.08 seconds.

Country	Avg Throughput	Avg Packet Loss
Algeria	455.4542094	2.882128
Bangladesh	724.8877068	1.283314
Bolivia	231.8753112	1.203456
Brazil	363.6881044	1.434288
Burkina Faso	159.7602764	13.10837
Canada	939.6410619	1.413246
China	12002.55675	1.193068
Denmark	807.3416842	3.115455
Germany	1063.38155	1.087691
Hungary	1391.708115	2.153679
India	316.9754833	2.342943
Italy	1185.884051	4.298317
Japan	424.941182	0.725209
Jordan	354.8952079	1.038937
Malaysia	1866.293044	1.914922
Nepal	3711.968379	1.194302
Pakistan	4423.471215	1.730353
Russia	345.0323558	6.203755
South Africa	364.4001095	3.290309
Sri Lanka	2979.726252	1.603051
Switzerland	3740.922352	1.1178
Taiwan	689.8856342	1.307937
UK	4315.385608	1.10725
United States	2029.08699	0.97399

Table. 1. Average throughput and packet loss of all PingER monitors between 1998 and 2014.

# V. CONCLUSION

This work presented an approach using a large-scale solution to deal with a huge amount of data in a data warehouse analytical environment. First, we used a dataflow mechanism capable of managing and executing a distributed ETL process to extract and transform Internet quality data from a huge dataset. Then, we created a data warehouse environment that showed to be quite efficient and capable of dealing with all the massive data from PingER project. Moreover, it showed to be very scalable, making the solution capable of dealing with even more data. The main advantage of the proposed approach is that it provides the facility to search the PingER database and filter on ways that the original PingER project was not capable of.

As future work, we highlight our intention to publish the queries results using Linked Open Data standards following the PingER LOD Ontology proposed by Souza *et al.* [16]. Our goal is to make the big data platform accessible to the general public, so researchers can access reliable data about the Internet quality around the world. Besides, we are already working on improving the dataflow to support all the necessary steps, from data extraction to data loading into an Impala table, using daily batch jobs (cron tabs) to automatize the ETL process and keep the data up-to-date.

#### ACKNOWLEDGEMENTS

T. M. S. Barbosa thanks, CAPES to the Scholarship Process 88888.042771/2013-00 through the Science without Borders Program. T. M. S. Barbosa is with CAPES Foundation, Ministry of Education of Brazil. M.L.M. Campos thanks FAPERJ (process number E-26/110.492/2012) and CNPq (process number 308934/2012-1). R. Souza also thanks CAPES for the partial support of this work. We especially thank all the team members from the GRECO and PESC Groups (Federal University of Rio de Janeiro) who helped in the work.

#### REFERENCES

- [1] Apache HDFS. (2015, Jul 20). HDFS Architecture Guide [Online]. Available: http://hadoop.apache.org/docs/r1.2.1/hdfs\_design.html.
- [2] Apache Hive. (2015, Jul 20). Apache Tajo: A big data warehouse system on Hadoop [Online]. Available: https://hive.apache.org/.
- [3] Apache Tajo. (2015, Jul 20). HDFS Architecture Guide [Online]. Available: http://tajo.apache.org/.
- [4] Cloudera. (2015, Jul 20). CDH Overview [Online]. Available: http://www.cloudera.com/content/cloudera/en/documentation/core/lates t/topics/cdh\_intro.html.
- [5] Cloudera. (2015, Jul 20). Cloudera Impala [Online]. Available: http://www.cloudera.com/content/cloudera/en/products-andservices/cdh/impala.html.
- [6] Cloudera. (2015, Jul 07). Tables [Online]. Available: http://www.cloudera.com/content/cloudera/en/documentation/clouderaimpala/latest/topics/impala\_tables.html.
- [7] Cloudera. (2015, Jul 07). Using the Parquet File Format with Impala Tables [Online]. Available: http://www.cloudera.com/content/cloudera/ en/documentation/cloudera-impala/latest /topics/impala\_parquet.html.
- [8] D. de Oliveira et al., "Debugging Scientific Workflows with Provenance: Achievements and Lessons Learned" in 29th SBBD – SBBD Proceedings, Curitiba, PR, Brazil, Oct 2014.
- [9] D. de Oliveira *et al.*, "SciCumulus: A Lightweigh Cloud Middleware to Explore Many Task Computing Paradigm in Scientific Workflows" in 2010 IEEE 3rd International Conference on Cloud Computing, 2010, pp. 378-385.
- [10] Database Research at Yale University. (2015, Jul 20). HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. [Online]. Available: http://db.cs.yale.edu/hadoopdb/hadoopdb.html.
- [11] Elastic. (2015, Oct 2). ElasticSearch [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/guide/current/intro.html
- [12] J. Dittrich et al. 'Efficient OR Hadoop: Why Not Both?" in Datenbank-Spektrum, Jan 2013.
- [13] J. Freire *et al.*, "Provenance for Computational Tasks: A Survey", Computing in Science & Engineering, vol.10, no. 3, pp. 11-21, May/June 2008, doi:10.1109/MCSE.2008.79
- [14] R. Kimball *et al.*, "The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses" John Wiley and Sons, 1998.
- [15] Red Hat. (2014, Dez 1). Tutorial on Internet Monitoring & PingER at SLAC [Online]. Available: http://www.slac.stanford.edu/comp/ net/wan-mon/tutorial.html.
- [16] R. Souza *et al.* "Linked Open Data Publication Strategies: Application in Networking Performance Measurement Data" Conference: 2014 ASE BigData, SocialCom, CyberSecurity Conference, Stanford University, CA, 2014.
- [17] R. L. Cottrell *et al.* (2015, Jul 07). Tutorial on Internet Monitoring & PingER at SLAC [Online]. Available: http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html.
- [18] W. Matthews and Les Cottrell, "The PingER Project: Active Internet Performance Monitoring for the HENP Community". Stanford University, IEEE Communications Magazine on Network Traffic Measurements and Experiments, May 2000.