Towards SDN-Enabled Big Data Platform for Social TV Analytics

Hu, Han; Wen, Yonggang; Gao, Yue; Chua, Tat-Seng; Li, Xuelong

2015

Hu, H., Wen, Y., Gao, Y., Chua, T.-S., & Li, X. (2015). Toward an SDN-enabled big data platform for social TV analytics. IEEE Network, 29(5), 43-49.

https://hdl.handle.net/10356/80956

https://doi.org/10.1109/MNET.2015.7293304

Towards SDN-Enabled Big Data Platform for Social TV Analytics

Han Hu, Yonggang Wen, Yue Gao, Tat-Seng Chua and Xuelong Li

Abstract

TV experience is being transformed with online social networks (OSNs). TV audience are sharing their opinions (i.e., social response) about video programs on OSNs (e.g., Twitter and Sina Weibo), thus providing a great opportunity for mining these data for stakeholders in TV value chains. This new paradigm is touted as **Social TV Analytics**, integrating the emerging big-data research into TV. In this article, we envision and develop a unified big-data platform for social TV analytics, extracting valuable insights from TV social response in a real-time manner. Such a platform presents tremendous challenges in networking architecture for our big-data platform. We propose to build a cloud-centric platform with software defined networking (SDN) support, providing on-demand virtual machines and reconfigurable network. The architecture of our system consists of three key components, including a robust data crawler system, an SDN enabled big data processing system, and a social media analytics system. The data crawler system adopts a distributed architecture to circumvent the access constraints of OSNs to crawl sufficient data about each TV program of interest; the SDN-enabled big data processing system integrates SDN and Hadoop, and exploits the SDN benefit to transfer intermediate data between different processing units to accelerate the data processing rate; and the social media analytics system extracts the public perception and knowledge related to TV programs based on microblog data. We have built a proof-of-concept demo over a private cloud at Nanyang Technological University (NTU). Feature verification and performance comparisons demonstrate the feasibility and effectiveness of the system.

Index Terms

Big Data, SDN, Hadoop, Social TV Analytics

I. Introduction

THE emergence of online social network (OSN) services enable users to share, socialize, and interact with other users easily. This generates a huge amount of social media data that can be analyzed to produce social analysis about users' relation, daily thought, comment, and concerns on various entities, such as the TV program. As reported by Nielsen [1], a third of active twitter users tweeted about TV-related content during June 2012, which refers to an increase of 27 percent from the beginning of that year. Mining the social media content associated with TV programs could bring a new business mode to the traditional TV ecosystem, offering targeted advertisement, interactive program composing, etc. Therefore, building a big data platform for social TV analytics has attracted more and more attention.

However, the nature of tweets, e.g., the limited length and the massive use of abbreviations, posits significant challenges to effectively collect, store, and analyze the vast amount of tweet messages. First, there are tons of tweets being posted every minute. It is difficult to elicit adequate TV programs related tweets in real-time under the access constraints set by most OSNs. Second, tweets are published continuously with complex data formats, including text, images, and social relationships. Big data platform needs to organize these diversity data in a convenient way and facilitate different levels of analysis, especially real-time analysis.

Collecting a large scale of high quality social media plays the paramount role in social networks related research and applications, such as political election prediction [2] and early warning of epidemics [3]. However, all data collection methods depend on the APIs (Application Programming Interfaces) provided by social media platforms, as well as the access constraints. In general, there are three main data collection strategies in social media, i.e., stream-based, user-based, and search-based. The stream-based methods rely on the platform to push posts according to a given set of keywords. The user-based strategies track the information related to specific users. The search-based methods query the given keywords to acquire posts. All these methods start from a pre-defined keyword/user set. Hence, the effectiveness of data collection methods depend on the completeness and representativeness of keyword/user set.

To effectively manage and process huge amount of tweets, several revolutionary technologies have been considered as the fundamental building blocks, including Hadoop [4] and SDN. The core component of Hadoop is the MapReduce [5] computing framework, involving two major tasks, Map and Reduce. Since MapReduce only provides a general paradigm to implement various applications, there are great opportunities for performance optimization, especially in network traffic control. For

H. Hu, Y. Gao and T.-S. Chua are with the School of Computing, National University of Singapore, Singapore 117417. e-mail: {huh, gaoy, chuats}@comp.nus.edu.sg. H. Hu is visiting NTU for this work.

Y. Wen is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798. e-mail: ygwen@ntu.edu.sg.

X. Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shanxi, P. R. China. e-mail: xuelong_li@opt.ac.cn.

Manuscript received December 1, 2013; revised January 11, 2014.

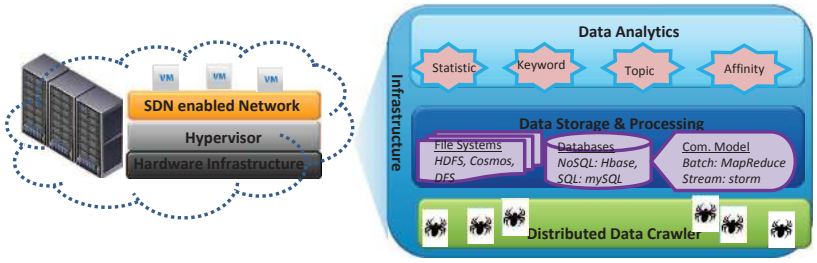


Fig. 1. A generic architecture for social TV analytics, consisting of four layers, i.e., infrastructure, distributed data crawler, data storage & processing, and data analytics

instance, the shuffle phase is a major component of task completion time [6]. Recently, the emergence of SDN provides a chance for more dynamic and flexible network operation. The core idea is to decouple the data plane and the control plane. Compared with the traditional network model, only the data plane resides on the switch, and the control logic is separately placed on logically centralized controllers. Through status query and flow table modification from the controllers, the application can dynamically sense and operate the underlying network to gain better performance. Wang et al. [7] exploited the centralized control feature of both SDN and Hadoop, and combined the SDN controller and Hadoop scheduler to achieve intelligent data flow routing.

Inspired by these research efforts, we focus on building a unified big data platform for social TV analytics. In particular, we design and implement a cloud-centric SDN enabled platform, which integrates the virtualization technology and the Openflow [8] tool together. Leveraging the platform, we customize our social TV analytics solution, involving three key components, i.e., a data crawler system, an SDN enabled big data processing system, and a social media analytics system. The data crawler system is deployed in a collection of nodes distributed in different IP segmentations. These nodes collaborate to crawl data from online social networks, while preventing proactively blocked by the requested sites. The SDN enabled big data process system is built on Hadoop, and exploits the network control with Openflow to accelerate the processing rate. The social media analytics system leverages the MapReduce paradigm to mine social response associated with TV programs.The unique contribution of this paper are the following:

- We propose a distributed data crawler platform that consists of two core components, including program descriptor and distributed crawler. Program descriptor describes a TV program by the keyword and key-user set, which can be dynamically enriched to guarantee the representativeness of the gathered microblog dataset. Distributed data crawler generates crawling tasks based on the keyword and key-user set and dispatches the tasks in a load-balance way to prevent proactively blocked by the OSNs.
- We implement a two-layer SDN enabled data analysis architecture based on Hadoop, which enables shuffling the intermediate data cross distributed data centers or computing clusters to accelerate the analysis rate.

Our prototype system has been implemented on top of a private cloud at NTU to demonstrate the concept and evaluate its performance. Some preliminary results are presented to illustrate the benefits of the SDN enabled big data processing platform, and deep analytics from tweet messages.

II. A Generic Big Data Platform for Social TV Analytics

A big-data system is complex, providing functions to deal with different phases in the digital data life cycle [9], including data generation phase, data acquisition phase, data storage phase, and data analytics phase . Following this functional requirement, we customize a generic big data platform for social TV analytics, as illustrated in Figure 6. The anatomy of this proposed architecture consists of four fundamental components, including an infrastructure layer, a distributed data crawler layer, a data storage and processing layer, and a data analytics layer, from a layered perspective. Each of them is elaborated as follows:

1) Infrastructure: By using the cloud computing paradigm, raw ICT resources, including CPU, storage, bandwidth, etc., are abstracted into a resource pool, and provided in the form of virtual machines (VMs). Furthermore, SDN enabled switches can be exploited to construct the data center network. The system administrator can monitor the utilization of VMs and network dynamics, and then dynamically adjust the network flow path via flow re-routing or bandwidth reservation. In this way, we can optimize the resource allocation and prevent the network congestion simultaneously.

2) Distributed Data Crawler: The distributed data crawler system is deployed in a collection of nodes distributed in several IP segmentations. One node will be elected as the scheduler to dispatch the crawling tasks to other crawler nodes. Each crawler node adopts multiple threads to crawl data. To prevent being blocked by OSNs, each node will occupy several application keys, which will be dynamically allocated to different threads. The goal of this layer is to crawl more data under the access limits set by OSNs.

3) Data Storage and Processing: The data storage and processing layer provides a unified scheme to effectively manage and process social media streams. In this work, we build our solution on two emerging big data analytics platforms, i.e., Hadoop and Storm. Hadoop integrates distributed file system (HDFS), NoSQL database (Hbase), and batch-style programming model (MapReduce). As a supplementary, Storm plays the role of streaming computing for real-time analysis.

4) Data Analytics: By leveraging the big data platform, data analytics aims to provide different level of analysis results, from statistics to content analysis, in term of social phenomena, sense, influences, etc., which helps to understand the social perception on TV programs. For example, who are watching a specific TV program, what is the background of the viewers, and what they are talking about. The whole TV ecosystem, from dramatists, TV producers, TV operators, to advertisement agencies, can benefit from this data analytics.

III. System Prototype

In this section, we highlight our social TV analytics system in the proposed generic system architecture by introducing its three key components, including a distributed data crawler component, an SDN enabled data processing component, and a TV program related social media analytics component.

A. Distributed Data Crawler

The distributed data crawler system aims to crawl tweets associated with TV programs effectively. It is the basis of the subsequent analysis phase. However, this is not a trivial task due to the following reasons: 1) The volume of TV programs related tweets is huge and we need to crawl a representative number of tweets to ensure coverage, relevance, and representativeness. 2) Most live microblog services set limits on the amount and frequency of data that can be acquired. Taking Sina Weibo, the largest micro-blogging platform with over 500 million users in China, as an example, the primary account is authorized to request 150 times per hour via the official APIs, and the response to an http query (corresponding to one keyword) only contains the first 50 pages, which accounts for 1000 tweets at most. Once you violate the rules, your belonging IP segmentation will be denied for access. To tackle these issues, we design the following strategies:

- *Program Descriptor:* In general, relevant tweets may contain keywords that have context relationship with the TV program, or be published by particular users, such as official accounts. Therefore, we design four types of items to describe a TV program, i.e., fixed keywords, dynamic keywords, known accounts, and dynamic key-users. Fixed keywords are first manually selected to uniquely identify a TV program. Based on the tweet set fixed keywords, we extract a list of temporally emerging terms as dynamic keywords. Similar to fixed keywords, known accounts are manually selected to identify a set of TV program related users, like directors' or actors' accounts. Dynamic key-users are those who post many related tweets and have many followers. These key-users are likely to post relevant messages which may not contain the known or evolving keywords during the time period of interest. The extraction method for dynamic keywords and key users will be elaborated in the data analytics section.
- *Distributed Crawler:* Given a set of keywords or key-users related to a TV program, we can use the items to query and collect tweets. In order to fully utilize every account and prevent being blocked by OSNs, the crawling procedure adopts a distributed manner. First, since different items correspond to distinct amounts of tweets, once the number of a query result exceeds the threshold, we will divide the query into a set of sub-queries. For example, we can constrain the query in a smaller time period. In this way, a regular query can be split into many sub-queries within a time slot or a region, like "keyword = k, time=0:00-1:00, region=r". Second, all the sub-queries are dispatched to crawler nodes distributed in different IP segmentations. Every crawler node adopts multi-thread to send requests. To prevent being blocked by OSNs, each thread will be dynamically combined with an application key. The thread and application key numbers are empirically determined according to the limits of access frequency and amount.

Figure 7 presents the architecture of our proposed distributed data crawler. Each TV program is depicted by a program descriptor, consisting of four items that can be dynamically expanded. For each item, our system will first send a query request to know how many tweets are relevant in the interested time period, and then decide whether and how we should split the query into several sub-queries. Following that, all the sub-queries are emitted to the task queue. Considering the access constraint, we implemented a resource pool to maintain the constraint related resources (named rare property in this paper), such as the application keys and IP address, etc. In addition, we exploited Zookeeper to monitor the running status of all machines. According to the system status, the scheduler will dispatch the tasks, as well as the rare property, to the execution nodes in a load-balance way. We have implemented two types of execution nodes, i.e., the API-agent and the html parser. The API-agent crawls tweets via the offered APIs, while the html parser extracts tweets directly from webpages. These two types of crawler nodes have their own advantages: the API-agent can acquire complex and complete information, while the html parser occupies less connection quota. When the allocated task is accomplished successfully, the execution node will notify the status, including the balance of the rare property, to the scheduler and the resource pool. Once Zookeeper detects that one of the nodes is down, the allocated task will be re-scheduled to another active node. Finally, all the tweets gathered are de-noised by an SVM classifier and then stored in our storage system.

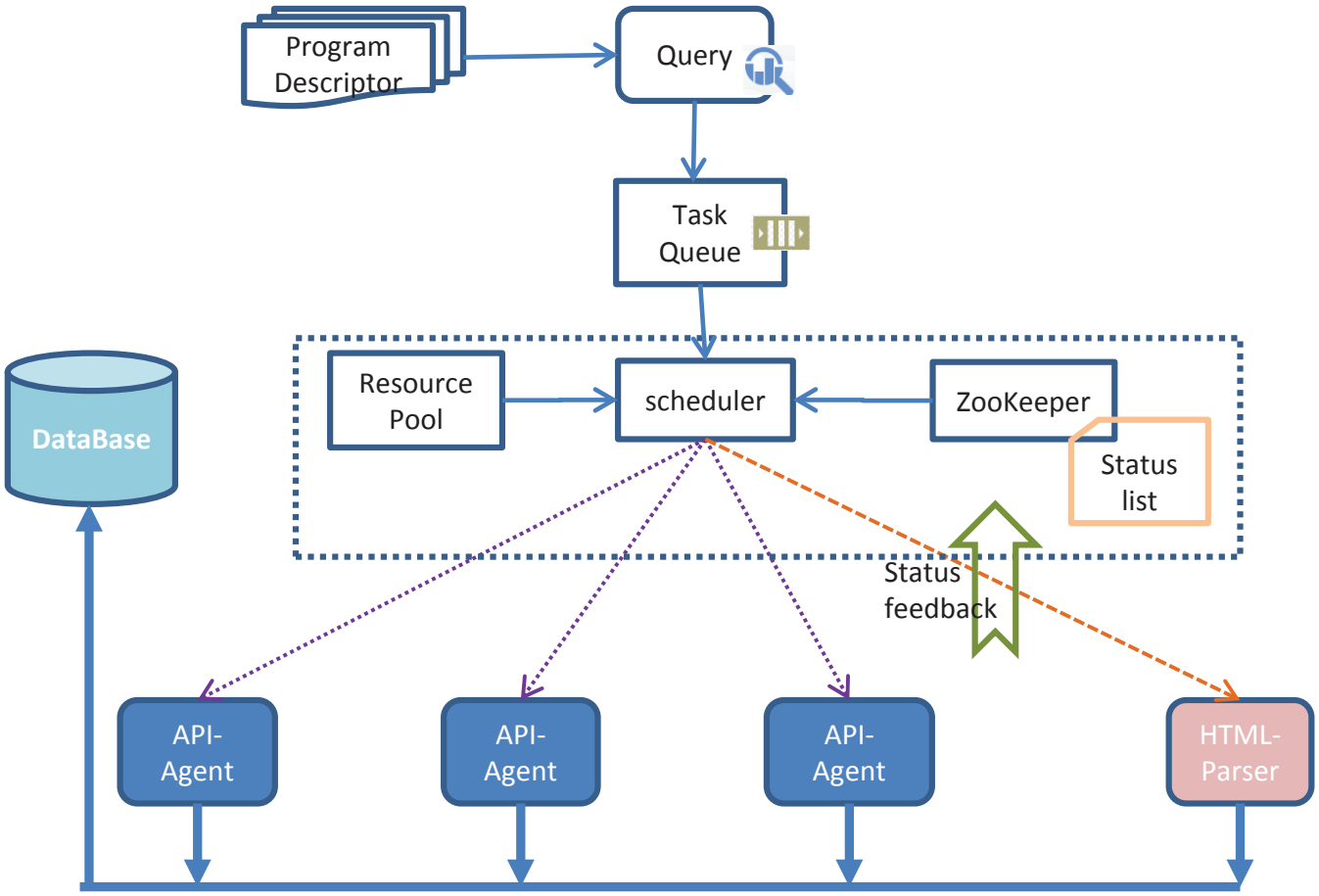


Fig. 2. Architecture of the distributed data crawler

B. SDN Enabled Data Processing Platform

As described in the previous section, our system exploits a set of nodes distributed in different IP segmentations to prevent being blocked by OSNs. Once we have collected an adequate number of tweets, we can merge them to a specific cluster and employ Hadoop for analysis. However, this method incurs significant network traffic and analysis-delay. An improved method is to analyze data locally, and merge the intermediate data to generate the final result. The benefit comes from two aspects: 1) in many cases, intermediate data is much smaller than the raw data; 2) network traffic in the shuffle stage is in waves, while sustained in the previous method, leading to higher possibility of network congestion. Unfortunately, current Hadoop does not support cross-site shuffle.

In this work, by leveraging the flow forwarding feature provided by SDN, we implemented an SDN enabled Hadoop framework to combine the SDN controller with the Hadoop job scheduler to tackle this issue, as depicted in Figure 8. The framework consists of two layers, i.e., a local layer and a global layer. The local layer comprises a collection of data centers located at different IP segmentations. The data center network is organized in a fat-tree or other network topologies using OpenFlow enabled switches. One centralized local controller is installed to monitor and configure the network flow, so that local shuffles can be moved to other data centers. The distributed data centers are interconnected via upper layer switches that are controlled by a logically centralized global controller. Local controller has the insight of local network traffic, while the global controller has the full view of network traffic among the data centers, enabling global traffic engineering. All local controllers are connected to global controller. Moreover, they can exchange and share network views.

On this platform, we can shuffle the intermediate data cross-sites. Since we need to transfer the intermediate data to a central data center, and then generate the final result. The execution time of this procedure consists of two parts, the transmission time of the intermediate data and the processing time in the central data center. Hence, the choice of the central data center will greatly affect the system performance. We solve this problem in two steps. First, all map tasks run the same map function. Hence, their intermediate data volume will be similar when the input size to the mappers is consistent. With the previous knowledge, the job scheduler can predict the traffic demand. Second, based on this estimation, we can calculate the approximate transmission time and processing time. The goal is to find a node that minimizes the sum of transmission time and processing time.

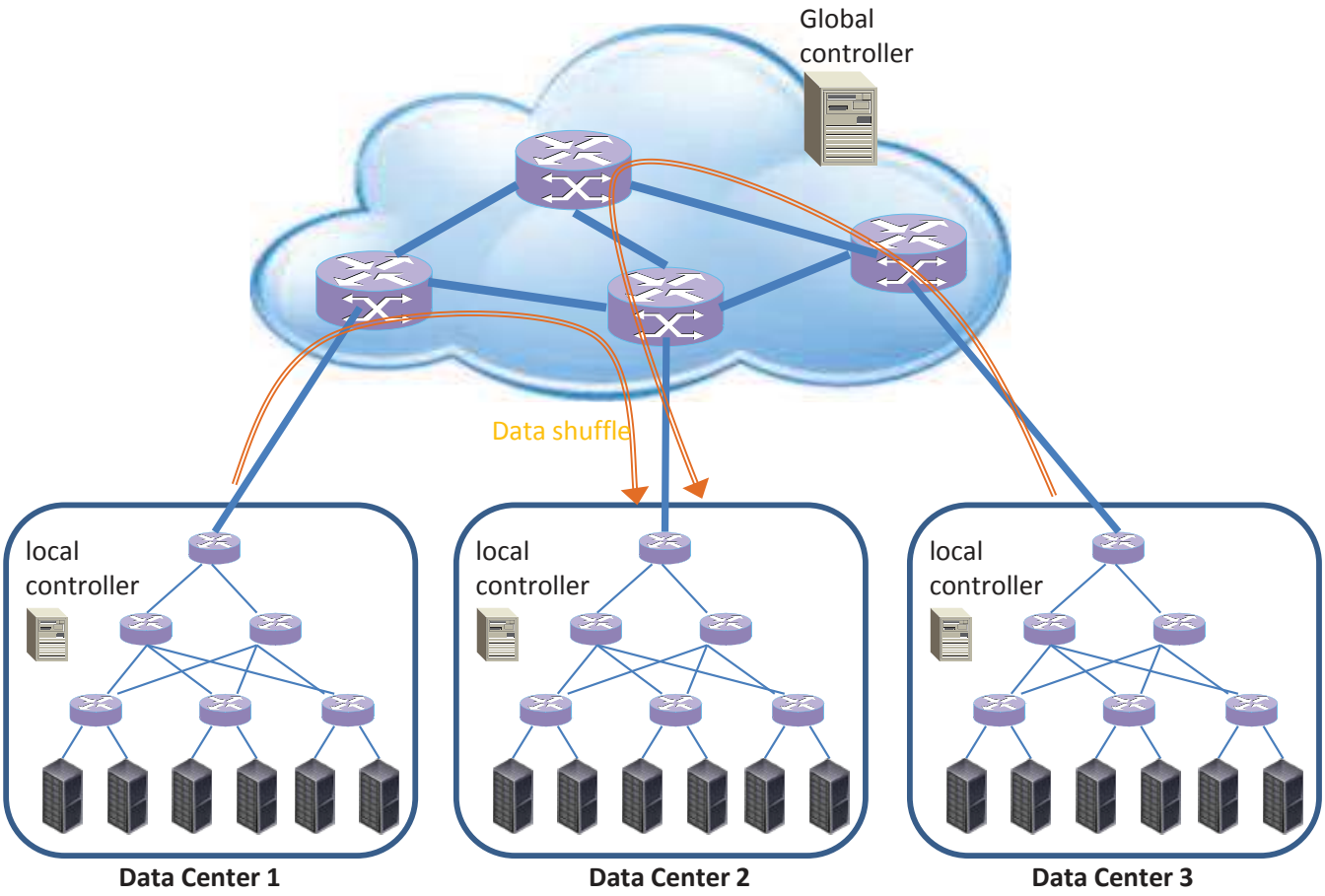


Fig. 3. Two-Layer architecture for SDN enabled Hadoop platform

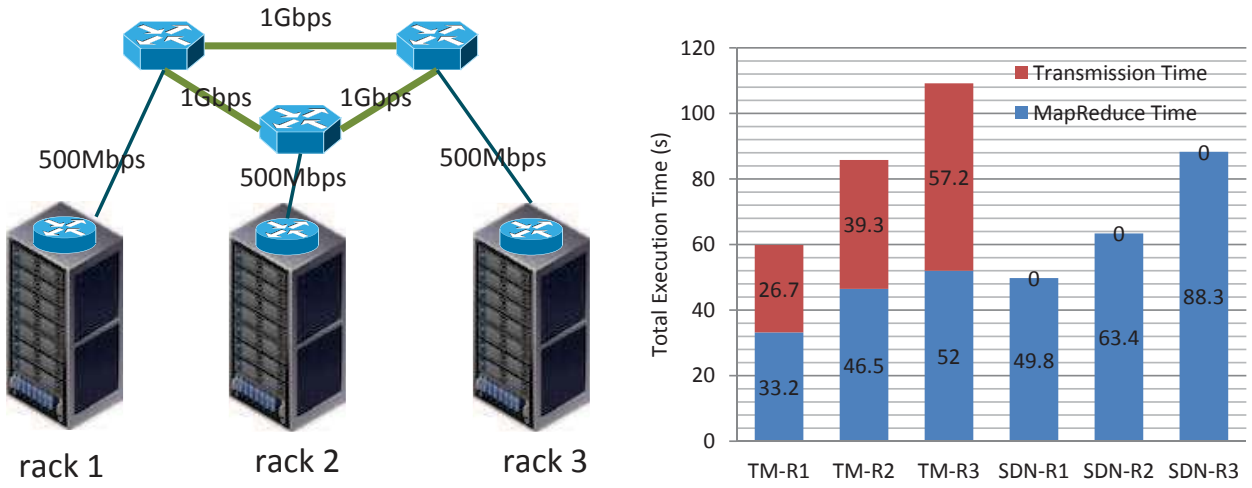
C. TV Program Related Social Media Analytics

Microblog services provide an essential platform for users to publish messages, which contain everyday thoughts, opinions, and experiences. Parts of these UGCs reflect their interests, concerns and criticisms about TV programs. The aim of social TV analytics is to associate the public perception with the TV program in realtime as the TV program is being broadcast. In general, microblog data contain images, text, and social relationships. Currently, we only focus on the text analysis, including statistics, keywords and key-users, and emerging topics. As for image analysis, there are many work [10][11][12][13] devoted in the past years, which can assist our future work.

We generate a collection of statistic metrics to reveal the public interest on a particular TV program, including tweets per hour/day, unique user number, shared by time, repost by time, geo-distribution, etc. These general metrics are beneficial to the whole TV ecosystem. For example, the metric geo-distribution can demonstrate the distribution of audience, and further assist targeted advertisement and program composing.

Keywords are words which occur in the tweets more often than we would expect to occur by chance alone. They can act as a reference for TV programs. Key-users are those who influence the information prorogation. Roughly speaking, we can know who are talking about what on TV programs from keyword and key-user sets. In the previous section, we set four types of items, fixed/dynamic keyword, and known/dynamic key-user. Dynamic keywords and key-users are extracted from the dataset identified by fixed keywords and known accounts. The generation of these two items is detailed as follows:

- *Dynamic Keyword Generation:* Considering a time slot t (e.g., one day), we can construct two tweets sets S_t and S_{t^-}, where S_t refers to the relevant tweets in the current time slot. While S_{t^-} covers all the tweets sent during the time period $[t-T, t]$, where T is the pre-defined time interval, e.g, one day or one week. The vocabulary sets of these two tweet sets are denoted as $W_t = \{w_1, w_2, ...\}$ and $W_{t^-} = \{w'_1, w'_2, ...\}$. We then identify the words that have different distributions in S_t and S_{t^-}. These words with rising frequencies are the potential emerging keywords.
- *Key User Generation:* At time slot t, given a time interval T, we get a user set U_t who posted at least one relevant tweets in the time window. For user $u_n \in U_t$, his activity is measured by the ratio of relevant tweets sent by himself during the time window. Considering the information propagation pattern, a user's score is calculated by incorporating the activity of all his followers. We then rank users in U_t by their score and the top N users are selected as the key users.



(a) Testbed architecture, consisting of three racks connected via three SDN switches

(b) Total execution time for two types of strategies, traditional method and SDN enabled architecture

Fig. 4. Performance evaluation for SDN enabled big data platform

As for topic modeling, we need to handle the live and large volume of tweets to detect the topics without any prior knowledge of the number of topics. We employ an online or incremental clustering algorithm to handle a constant stream of new tweets. In particular, according to the similarity between any tweets, all the tweets will be categorized to different clusters, representing topics. Finally we analyze the emerging topic-related features, including user authority, tweets influence, and emerging keywords. These features are incorporated into a topic learner to identify the emerging topics on the fly [14].

IV. Evaluation

In this section, we first describe our testbed built on a private data center. Following that, we conduct a performance comparison to demonstrate the efficiency of our SDN-enabled big data platform. Finally, we present some preliminary results on social TV analytics.

A. System Setup

We built our system on top a modular data center at NTU, which consists of 10 racks. Each rack contains up to 30 HP servers and 2 Gigabit Cisco switches. The data center can provide an ICT capacity of 25 TB disk, 1200 GB memory, and 600 CPUs. We utilized CloudStack to virtualize physical machines into a collection of virtual machines in the infrastructure layer.

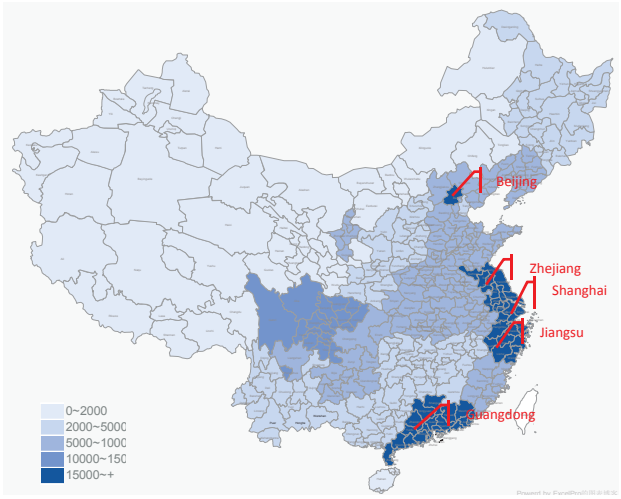
B. Performance Evaluation for SDN Enabled Big Data Platform

In order to evaluate the performance of our big data platform, we employed 3 racks to build a two-layer architecture, as shown in Figure 9(a). The numbers of servers in 3 racks are 10, 8 and 5, respectively. All servers in a rack are connected to the corresponding SDN switch via the ToR switch with the link capacity of 500 Mbps. We configured three racks into different IP segmentations. The link capacity between two SDN switches is 1 Gbps. The Java-based OpenFlow Controller, Floodlight, is installed to setup flow entries in the OpenFlow switches. We used the Hadoop Sort program to run as the job under test. Each server has to process 5 GB tweets. We compared two types of solutions, i.e., traditional method (TM-) and SDN scheme (SDN-). For each type of solutions, we could either move the raw data to one of the three racks to process (marked as TM-R1, TM-R2 or TM-R3), or only shuffle the intermediate data (marked as SDN-R1, SDN-R2, or SDN-R3).

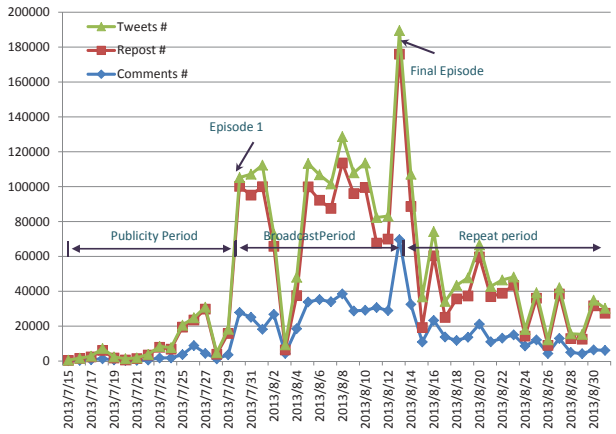
Figure 9(b) shows the execution time for different methods. When we used the same rack to process the intermediate data or raw data, our scheme can finish the task with less time. Moreover, the shortest execution time of our scheme is much less than that of the traditional methods. In our experimental configuration, the processing capability of rack 1 is larger than the other two racks, hence moving the data to rack 1 will lead to less execution time.

C. Social Perception for TV Programs

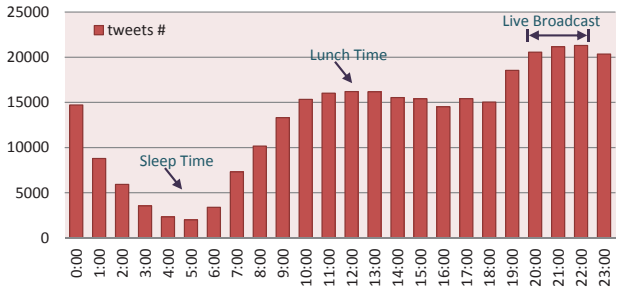
Social TV analytics aims to discover the social perception on TV programs in the context of social media. Such information is determined by statistical analysis and content analysis. In this paper, we provide several preliminary results in term of tweets geographical distribution, tweets number in different days, tweets time distribution, and keyword cloud, as illustrated in Figure 10. In this experiment, we take “Longmen Express”, a hot TV program played from 31/Jul/2013 to 14/Aug/2013, as demonstration. We crawled 315, 337 tweets from Sina Weibo, involving 240, 465 unique users. Figure 10(a) shows the



(a) Tweets geographical distribution



(b) Tweets number in different days



(c) Tweets time distribution

(d) Keyword cloud

Fig. 5. Social perception on TV programs

geographical distribution of our dataset. According to the color depth, we can see that audience in Guangdong, Beijing and Shanghai published more tweets than that published by viewers in other regions. Figure 10(b) presents the degree of interest over time, including the number of tweets, reposts, and comment respectively. During the broadcast period, audience posted more tweets on this program, and cumulated to a peak during the airing of the final episode. Figure 10(c) reveals the habits of audience posting microblogs. During the live broadcast period, audience are accustomed to watch TV and publish their opinions simultaneously. Figure 10(d) extracts the keywords aggregated from all the tweets.

V. Conclusion

In this paper, we proposed a cloud-centric big data platform for social TV analytics, aiming to mine social perception on TV programs from social media. Our system consists of three key components, i.e., the distributed data crawler, the big data processing and the social TV analytics. The distributed data crawler collects the tweets from online social networks, and then stores them in the big data platform. Using the big data processing paradigm, the social TV analytics implements different levels of analysis on the collected tweets. A proof-of-concept demo has been built on top of a private cloud at NTU. Feature verification and performance comparison demonstrate the feasibility and effectiveness of our innovative system. For future work, we will further study the affinity between TV audience and brands, social media assisted video tagging.

References

[1] Nielsen, "State of the media: The social media reprot," http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2012-Reports/The-Social-Media-Report-2012.pdf, 2012.

[2] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment." *ICWSM*, vol. 10, pp. 178–185, 2010.

[3] A. Culotta, "Towards detecting influenza epidemics by analyzing twitter messages," in *Proceedings of the first workshop on social media analytics*. ACM, 2010, pp. 115–122.

[4] T. White, *Hadoop: The definitive guide*. O'Reilly Media, Inc., 2012.
[5] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[6] M. Hammoud, M. S. Rehman, and M. F. Sakr, "Center-of-gravity reduce task scheduling to lower mapreduce network traffic," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 2012, pp. 49–58.
[7] G. Wang, T. Ng, and A. Shaikh, "Programming your network at run-time for big data applications," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 103–108.
[8] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
[9] F. Gallagher, "The big data value chain." [Online]. Available: http://fraysen.blogspot.sg/2012/06/big-data-value-chain.html.
[10] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 260–274, 2009.
[11] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Nenmf: an optimal gradient method for nonnegative matrix factorization," *Signal Processing, IEEE Transactions on*, vol. 60, no. 6, pp. 2882–2898, 2012.
[12] N. Guan, D. Tao, and Z. Luo, "Online nonnegative matrix factorization with robust stochastic approximation," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 1087–1099, 2012.
[13] T. Zhou and T. Dacheng, "Unmixing Incoherent Structures of Big Data by Randomized or Greedy Decomposition," 2013. [Online]. Available: http://arxiv.org/abs/1309.0302
[14] Y. Chen, H. Amiri, Z. Li, and T.-S. Chua, "Emerging topic detection for organizations from microblogs," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013, pp. 43–52.

Han Hu is a research fellow with the School of Computing at the National University of Singapore. His research interests include social media distribution and analysis. Hu has a PhD in control theory and control engineering from the Department of Automation at the University of Science and Technology of China, Hefei, China.

Yonggang Wen is an assistant professor in the School of Computer Engineering at the Nanyang Technological University (NTU), Singapore. His research interests include cloud computing, content networking, and green networks. Wen has a PhD in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT).

Yue Gao is a senior research fellow with the School of Computing at the National University of Singapore. His current research interests include multimedia information retrieval, especially image search and 3D object retrieval, and social media analysis. Gao has a PhD in control theory and control engineering from the Department of Automation at control theory and control engineering from the Department of Automation at Tsinghua University.

Tat-Seng Chua is a full professor with the School of Computing at the National University of Singapore. His research interests include multimedia information processing, including the extraction, retrieval, and questioning and answering of text and video information. Chua has a PhD in computer science and technology from the University of Leeds, UK.

Xuelong Li (M'02-SM'07-F'12) is a full professor with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics at the Chinese Academy of Sciences. His research interests include visual surveillance, biometrics, data mining, multimedia information retrieval, and industrial applications. Li has a PhD in signal and information processing from the University of Science and Technology of China. He is a fellow of IEEE.

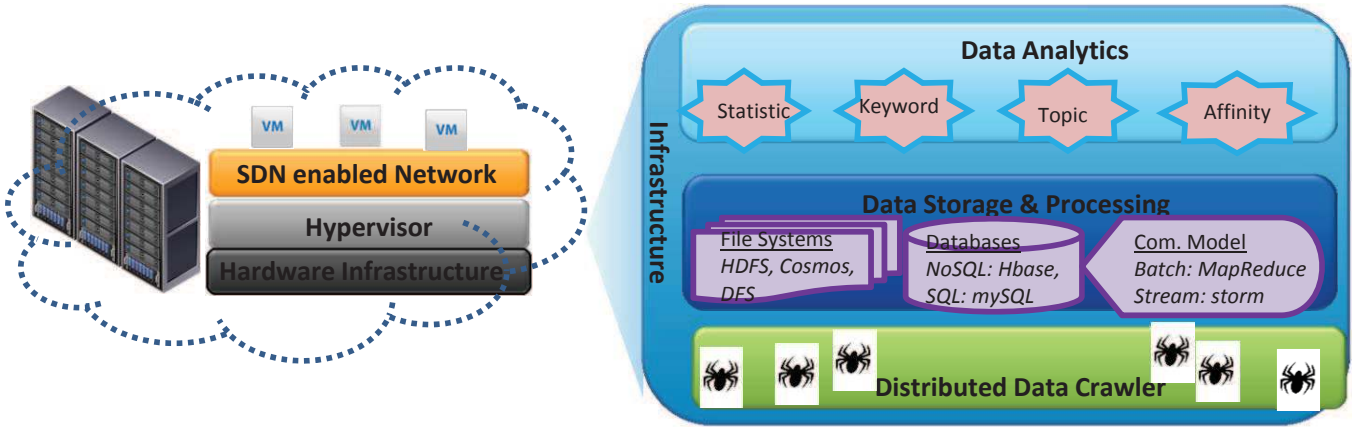


Fig. 6. A generic architecture for social TV analytics, consisting of four layers, i.e., infrastructure, distributed data crawler, data storage & processing, and data analytics

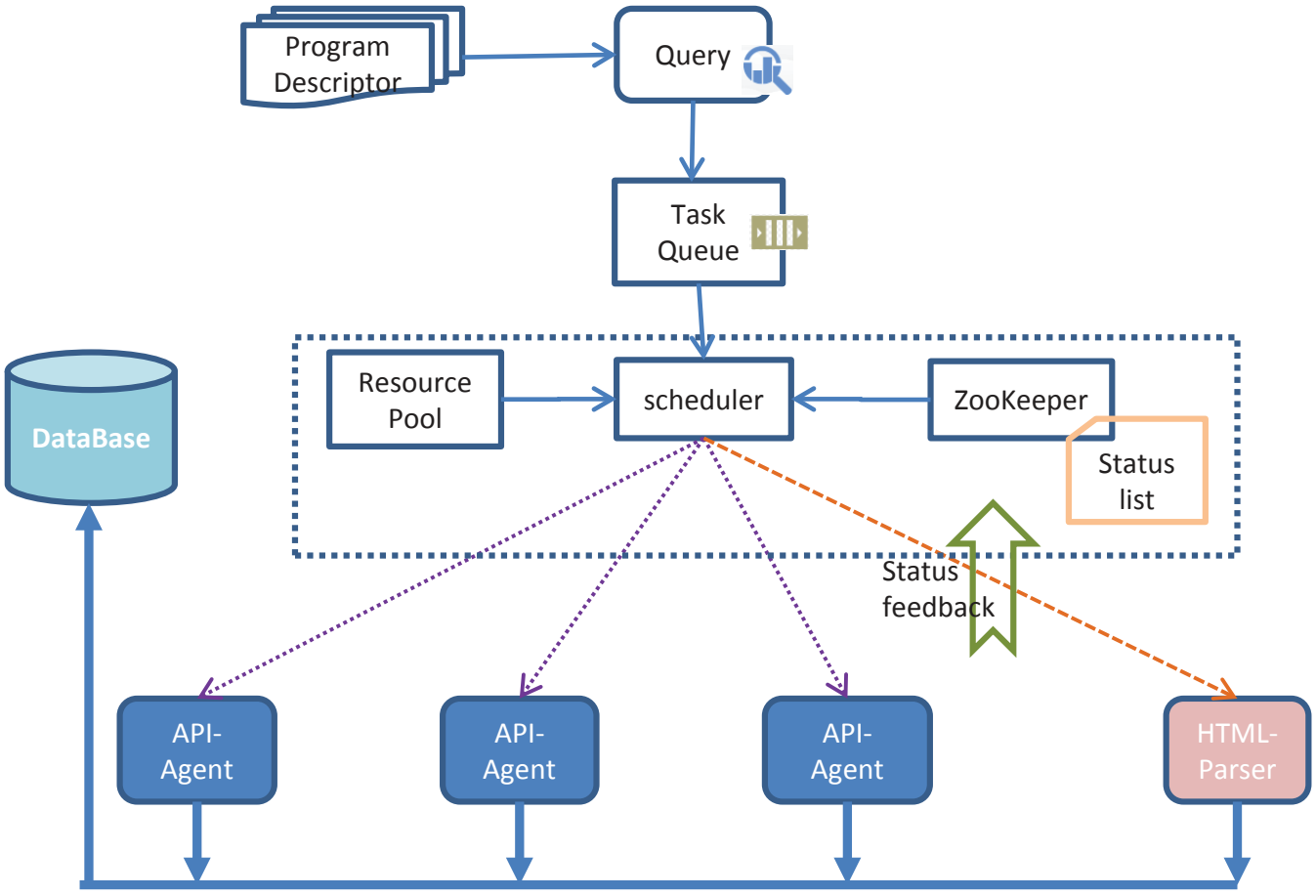


Fig. 7. Architecture of the distributed data crawler

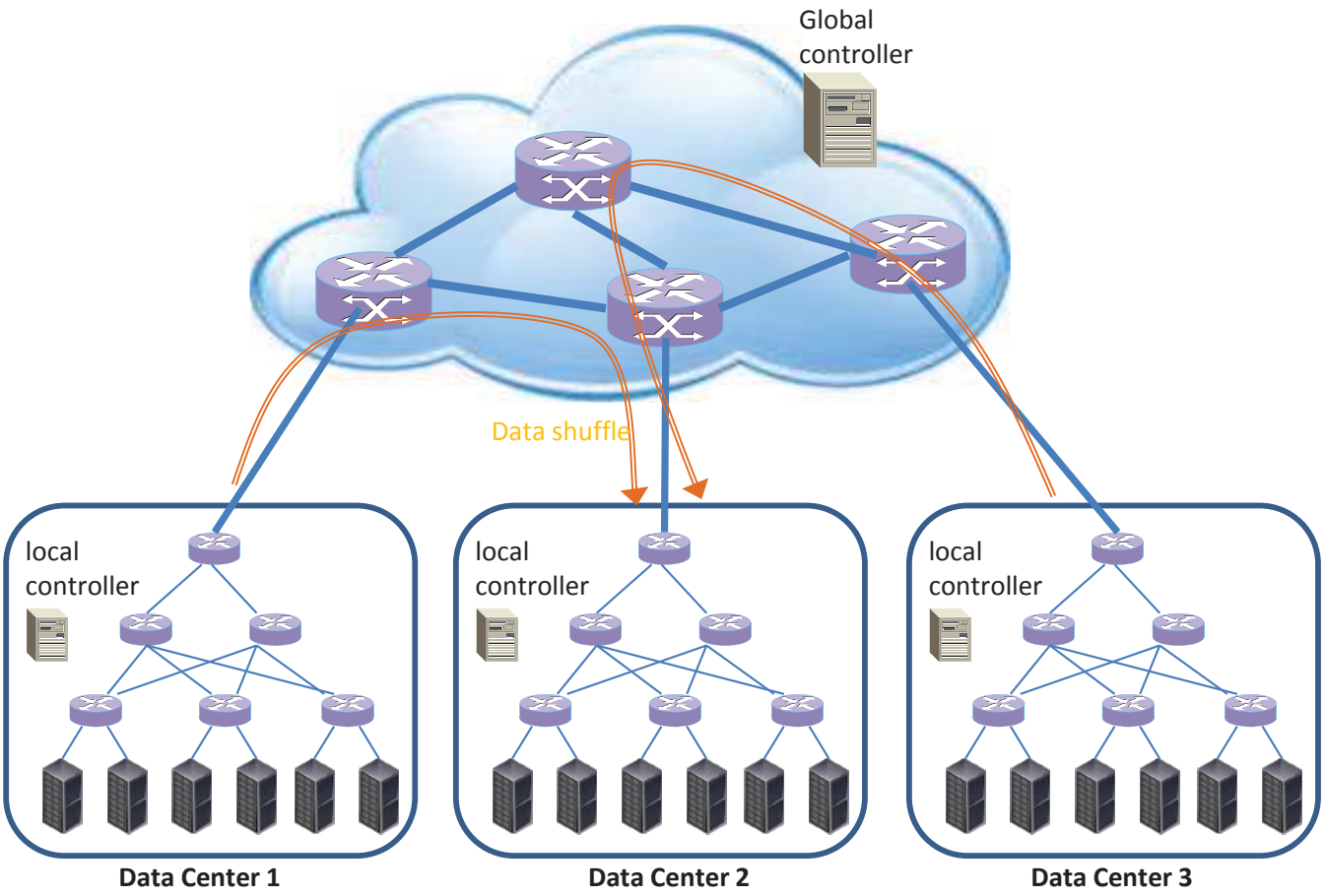
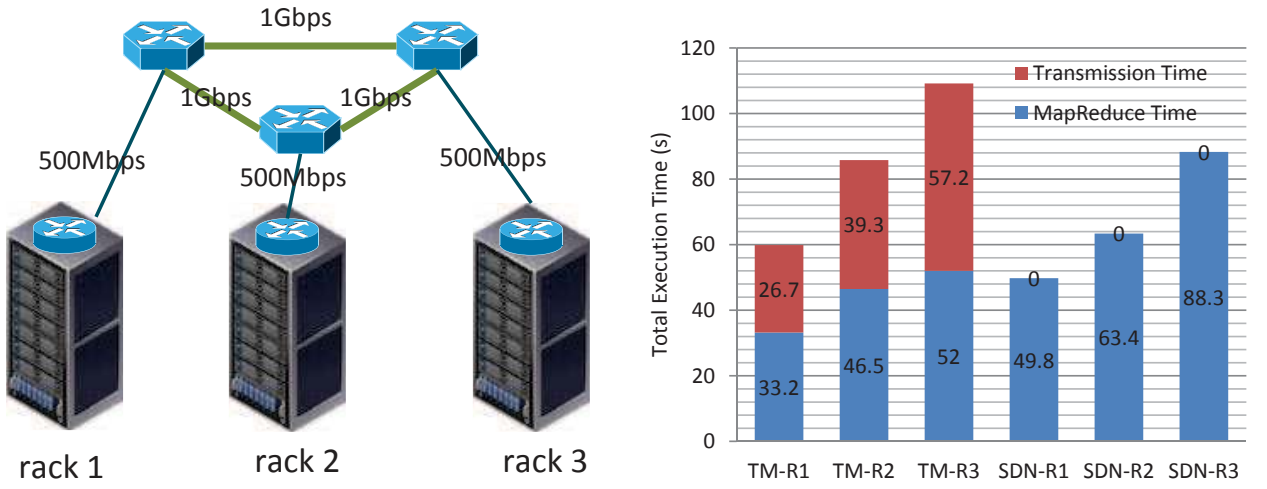


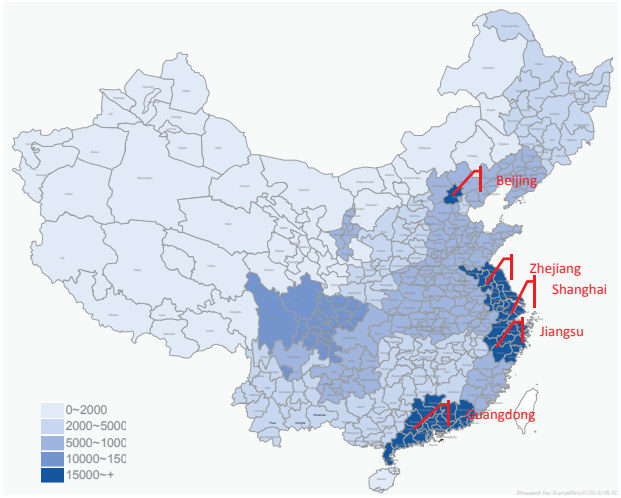
Fig. 8. Two-Layer architecture for SDN enabled Hadoop platform



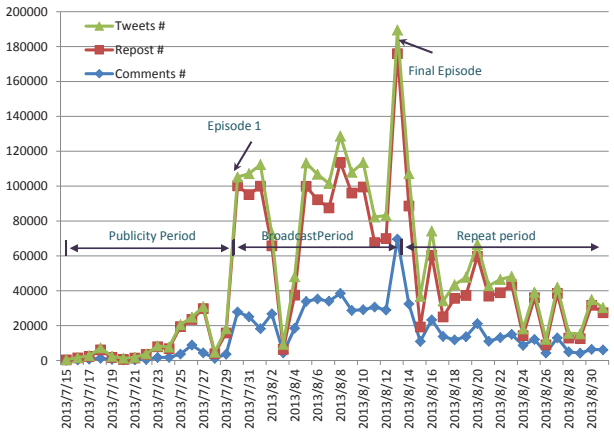
(a) Testbed architecture, consisting of three racks connected via three SDN switches

(b) Total execution time for two types of strategies, traditional method and SDN enabled architecture

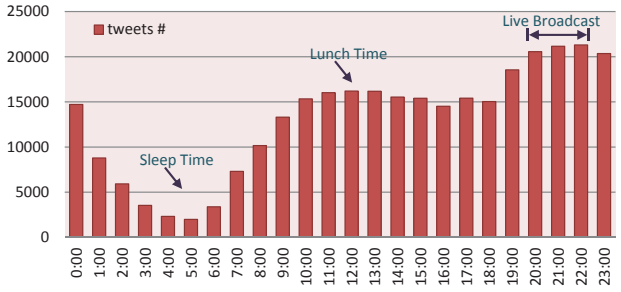
Fig. 9. Performance evaluation for SDN enabled big data platform



(a) Tweets geographical distribution



(b) Tweets number in different days



(c) Tweets time distribution

(d) Keyword cloud

Fig. 10. Social perception on TV programs