ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies

AN ADAPTABLE AND PREDICTABLE MATCHING SERVICE FOR CONTENT-BASED PUBLISHING DATA SYSTEMS

T.Nagendra Babu[*1] and M.Purushottam[2]
[*1]M.Tech (CSE) Student, SEAGI -Tirupati
[2]Assistant Professor, SEAGI -Tirupati

ABSTRACT

Characterized by the increasing arrival rate of live content, the emergency applications pose a great challenge: how to disseminate large-scale live content to interested users in a scalable and reliable manner. The publish/subscribe (pub/sub) model is widely used for data dissemination because of its capacity of seamlessly expanding the system to massive size. However, most event matching services of existing pub/sub systems either lead to low matching throughput when matching a large number of skewed subscriptions, or interrupt dissemination when a large number of servers fail. The cloud computing provides great opportunities for the requirements of complex computing and reliable communication. In this paper, we propose SREM, a scalable and reliable event matching service for content-based pub/sub systems in cloud computing environment. To achieve low routing latency and reliable links among servers, we propose a distributed overlay Skip Cloud to organize servers of SREM. Through a hybrid space partitioning technique HPartition, large-scale skewed subscriptions are mapped into multiple subspaces, which ensures high matching throughput and provides multiple candidate servers for each event. Moreover, a series of dynamics maintenance mechanisms are extensively studied. To evaluate the performance of SREM, 64 servers are deployed and millions of live content items are tested in a Cloud Stack test bed.

Keywords:-*Data systems, Content based publishing.*

I. INTRODUCTION

Because of the importance in helping users to make real time decisions, data dissemination has become dramatically significant in many large-scale emergency applications, such as earthquake monitoring, disaster weather warning, and status update in social networks. Recently, data dissemination in these emergency applications presents a number of fresh trends. One is the rapid growth of live content. For instance, Facebook users publish over 600,000 pieces of content and Twitter users send over 100,000 tweets on average per minute. The other is the highly dynamic network environment. For instance, the measurement studies indicates that most users' sessions in social networks only last several minutes. In emergency scenarios, the sudden disasters like earthquake or bad weather may lead to the failure of a large number of users instantaneously. These characteristics require the data dissemination system to be scalable and reliable. Firstly, the system must be scalable to support the large amount of live content. The key is to offer a scalable event matching service to filter out irrelevant users. Otherwise, the content may have to traverse a large number of uninterested users before they reach interested users. Secondly, with the dynamic network environment, it's quite necessary to provide reliable schemes to keep continuous data dissemination capacity. Otherwise, the system interruption may cause the live content becomes obsolete content.

Driven by these requirements, publish/subscribe (pub/sub) pattern is widely used to disseminate data due to its flexibility, scalability, and efficient support of complex event processing. In pub/sub systems (pub/subs), a receiver (subscriber) registers its interest in the form of a subscription. Events are published by senders to the pub/sub system. The system matches events against subscriptions and disseminates them to interested subscribers. In traditional data dissemination applications, the live content are generated by publishers at a low speed, which makes many pub/subs adopt the multi-hop routing techniques to disseminate events. A large body of broker-based pub/subs forward events and subscriptions through organizing nodes into diverse distributed overlays, such as tree based design, cluster-based design and DHT-based design. However, the multihop routing techniques in these broker-based systems lead to a low matching throughput, which is inadequate to apply to current high arrival rate of live content. Recently, cloud computing provides great opportunities for the applications of complex computing and high speed communication, where the servers are connected by high speed networks, and have powerful computing and storage capacities. A number of pub/sub services based on the cloud computing environment have been proposed, such as Move, BlueDove and SEMAS. However, most of them can not completely meet the requirements of both scalability and reliability when matching large-scale live content under highly dynamic environments. This mainly stems from the following facts:

ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies

1) Most of them are inappropriate to the matching of live content with high data dimensionality due to the limitation of their subscription space partitioning techniques, which bring either low matching throughput or high memory overhead.

2) These systems adopt the one-hop lookup technique among servers to reduce routing latency. In spite of its high efficiency, it requires each dispatching server to have the same view of matching servers.

Otherwise, the subscriptions or events may be assigned to the wrong matching servers, which brings the availability problem in the face of current joining or crash of matching servers. A number of schemes can be used to keep the consistent view, like periodically sending heartbeat messages to dispatching servers or exchanging messages among matching servers. However, these extra schemes may bring a large traffic overhead or the interruption of event matching service. Motivated by these factors, we propose a scalable and reliable matching service for content-based pub/sub service in cloud computing environments, called SREM. Specifically, we mainly focus on two problems: one is how to organize servers in the cloud computing environment to achieve scalable and reliable routing. The other is how to manage subscriptions and events to achieve parallel matching among these servers. Generally speaking, we provide the following contributions:

_ We propose a distributed overlay protocol, called Skip Cloud, to organize servers in the cloud computing environment. SkipCloud enables subscriptions and events to be forwarded among brokers in a scalable and reliable manner. Also it is easy to implement and maintain.

_To achieve scalable and reliable event matching among multiple servers, we propose a hybrid multidimensional space partitioning technique, called HPartition. It allows similar subscriptions to be divided into the same server and provides multiple candidate matching servers for each event. Moreover, it adaptively alleviates hot spots and keeps workload balance among all servers. We implement extensive experiments based on a CloudStack test bed to verify the performance of SREM under various parameter settings.

II. LITURATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system.

UNIVERSAL CROSS-CLOUD COMMUNICATION

Integration of applications, data-centers, and programming abstractions in the cloud-of-clouds poses many challenges to system engineers. Different cloud providers offer different communication abstractions, and applications exhibit different communication patterns. By abstracting from hardware addresses and lower-level communication, the publish/subscribe paradigm seems like an adequate abstraction for supporting communication across clouds, as it supports many-to-many communication between publishers and subscribers, of which one-to-one or one-to-many can be viewed as special cases. In particular, content-based publish/subscribe (CPS) systems provide an expressive abstraction that matches well with the key-value pair model of many established cloud storage and computing systems, and decentralized overlay-based CPS implementations scale up well. However, CPS systems perform poorly at small scale, e.g., one-to-one or one-to-many communication. This holds especially for multi-send scenarios which we refer to as entourages that may range from a channel between a publisher and a single subscriber to a broadcast between a publisher and a handful of subscribers. These scenarios are common in cloud computing, where cheap hardware is exploited for parallelism (efficiency) and redundancy (fault-tolerance). With CPS, multi-send messages go over several hops before their destinations are even identified via predicate matching, resulting in increased latency, especially when destinations are located in different data-centers or zones.Topic-based publish/subscribe (TPS) systems support communication at small scale more efficiently, but still route messages over multiple hops and inversely lack the flexibility of CPS systems. In this paper, we propose CPS protocols for cloud-of-clouds communication that can dynamically identify entourages of publishers and corresponding subscribers. Our CPS protocols dynamically connect the publishers with their entourages throughüberlays . These überlays can transmit messages from a publisher to its corresponding subscribers with low latency. Our experiments show that our protocols make CPS abstraction viable and beneficial for many applications. We introduce a CPS system named Atmosphere that leverages out CPS protocols and illustrate how Atmosphere has allowed us to implement, with little effort, versions of the popular HDFS and ZooKeeper systems which operate efficiently across data-centers.

ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies

MOVE: A LARGE SCALE KEYWORD-BASED CONTENT FILTERING AND DISSEMINATION SYSTEM

The Web 2.0 era is characterized by the emergence of a very large amount of live content. A real time and fine grained content filtering approach can precisely keep users up-to-date the information that they are interested. The key of the approach is to offer a scalable match algorithm. One might treat the content match as a special kind of content search, and resort to the classic algorithm [5]. However, due to blind flooding, [5] cannot be simply adapted for scalable content match. To increase the throughput of scalable match, we propose an adaptive approach to allocate (i.e, replicate and partition) filters. The allocation is based on our observation on real datasets: most users prefer to use short queries, consisting of around 2-3 terms per query, and web content typically contains tens and even thousands of terms per article. Thus, by reducing the number of processed documents, we can reduce the latency of matching large articles with filters, and have chance to achieve higher throughput. We implement our approach on an open source project, Apache Cassandra. The experiment with real datasets shows that our approach can achieve around folds of better throughput than two counterpart state-of-the-arts solutions.

A SCALABLE AND ELASTIC PUBLISH/SUBSCRIBE SERVICE

The rapid growth of sense-and-respond applications and the emerging cloud computing model present a new challenge: providing publish/subscribe as a scalable and elastic cloud service. This paper presents the Blue Dove attribute based publish/subscribe service that seeks to address such a challenge. Blue Dove uses a gossip-based one-hop overlay to organize servers into a scalable cluster. It proactively exploits skewness in data distribution to achieve high performance. By assigning each subscription to multiple servers through a multidimensional subscription space partitioning technique, it provides multiple candidate servers for each publication message. A message can be matched on any of its candidate servers with one hop forwarding. The performance-aware forwarding in Blue Dove ensures that the message is sent to the least loaded candidate server for processing, leading to low latency and high throughput. The evaluation shows that Blue Dove has a linear capacity increase as the system scales up, adapts to sudden workload changes within tens of seconds, and achieves multifold higher throughput than the techniques used in the existing enterprise and peer-to-peer pub/sub systems.

III. SYSTEM ANALYSIS

EXISTING SYSTEM:

- In traditional data dissemination applications, the live content are generated by publishers at a low speed, which makes many pub/subs adopt the multi-hop routing techniques to disseminate events.
- A large body of broker-based pub/subs forward events and subscriptions through organizing nodes into diverse distributed overlays, such as tree based design, cluster-based design and DHT-based design.

DISADVANTAGES OF EXISTING SYSTEM:

- The system cannot scalable to support the large amount of live content.
- The MultiHop routing techniques in these broker-based systems lead to a low matching throughput, which is inadequate to apply to current high arrival rate of live content.
- Most of them are inappropriate to the matching of live content with high data dimensionality due to the limitation of their subscription space partitioning techniques, which bring either low matching throughput or high memory overhead.

PROPOSED SYSTEM:

- Specifically, we mainly focus on two problems: one is how to organize servers in the cloud computing environment to achieve scalable and reliable routing. The other is how to manage subscriptions and events to achieve parallel matching among these servers.
- We propose a distributed overlay protocol, called Skip Cloud, to organize servers in the cloud computing environment. Skip Cloud enables subscriptions and events to be forwarded among brokers in a scalable and reliable manner. Also it is easy to implement and maintain.
- To achieve scalable and reliable event matching among multiple servers, we propose a hybrid multidimensional space partitioning technique, called HPartition. It allows similar subscriptions to be

ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies

divided into the same server and provides multiple candidate matching servers for each event. Moreover, it adaptively alleviates hot spots and keeps workload balance among all servers.

ADVANTAGES OF PROPOSED SYSTEM:

- ✓ We propose a scalable and reliable matching service for content-based pub/sub service in cloud computing environments, called SREM.
- ✓ We propose a hybrid multidimensional space partitioning technique, called HPartition SSPartition.

To alleviate the hot spots whose subscriptions fall into a narrow space, we propose a subscription set partitioning.

IV. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Modules

- HPartition
- Subscription Installation
- Event Assignment and Forwarding Strategy

V. MODULE DESCRIPTION

HPartition

In order to take advantage of multiple distributed brokers, SREM divides the entire content space among the top clusters of SkipCloud, so that each top cluster only handles a subset of the entire space and searches a small number of candidate subscriptions. SREM employs a hybrid multidimensional space partitioning technique, called HPartition, to achieve scalable and reliable event matching. Generally speaking, HPartition divides the entire content space into disjoint subspaces. Subscriptions and events with overlapping subspaces are dispatched and matched on the same top cluster of SkipCloud. To keep workload balance among servers, HPartition divides the hot spots into multiple cold spots in an adaptive manner.

Subscription Installation

A user can specify a subscription by defining the ranges over one or more dimensions. When receiving a subscription S, the broker first obtains all subspaces which are overlapping with S. Based on the logical space construction of HPartition, all dimensions are classified into t individual spaces. For the subscription S, its every matched subspace is hashed to a b-ary identifier with length of m. Thus, S is forwarded to the top cluster whose ClusterID is nearest to subspace hashed value. When a broker in the top cluster receives the subscription S ,it broadcasts S to other brokers in the same cluster, such that the top cluster provides reliable event matching and balanced workloads among its brokers.

Event Assignment and Forwarding Strategy

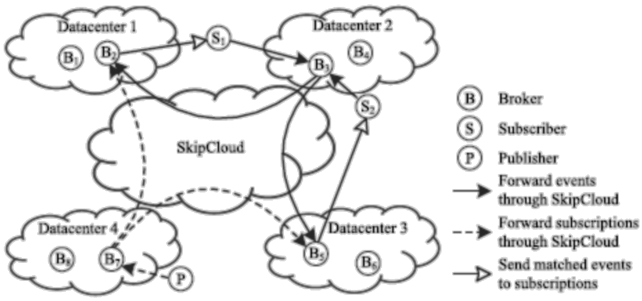
Upon receiving an event, the broker forwards this event to its corresponding subspaces.
HPartition divides the whole space into t separated spaces, which indicates there are t candidate subspaces for each event. Because of the different workloads, the strategy of how to select one candidate subspace for each event may greatly affect the matching rate. For an event e, suppose each candidate subspace contains N subscriptions. In HPartition, we adopt a probability based forwarding strategy to dispatch events. Therefore, this approach has better workload balance than the least first strategy and less searching latency than the random strategy.

ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies

SYSTEM ARCHITECTURE:



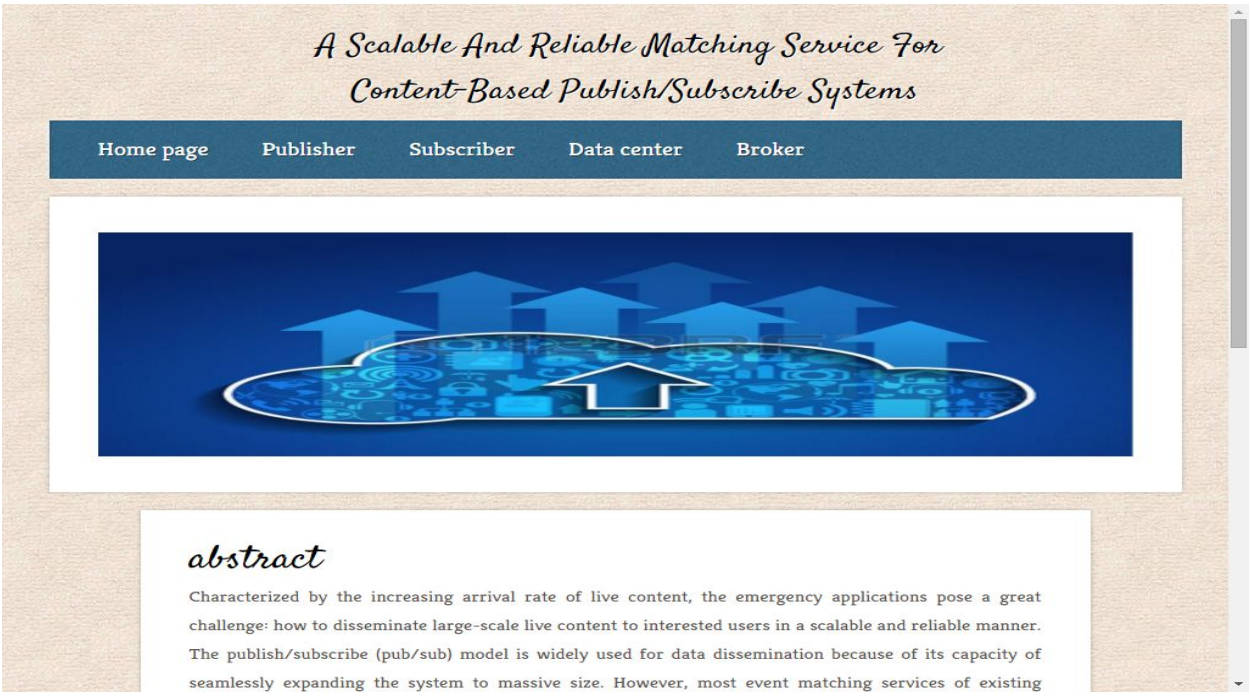
HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Ram :2GB

SOFTWARE REQUIREMENTS:

- Operating system : Windows XP/7.
- Coding Language: JAVA/J2EE
- IDE : Netbeans 7.4
- Database : MYSQL

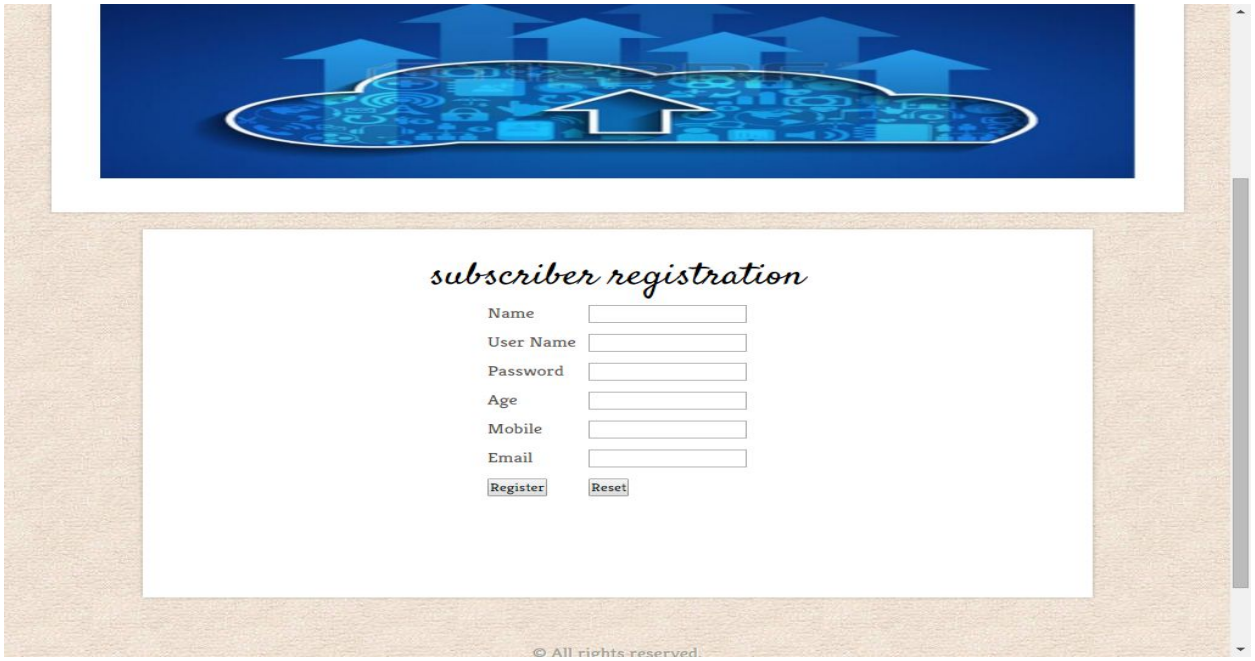
VI. RESULT



ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

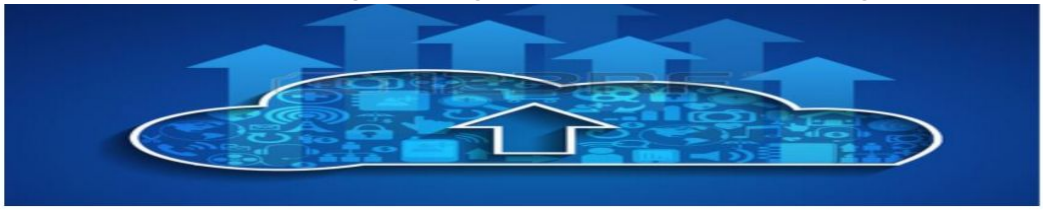
International Journal of Engineering Researches and Management Studies



ISSN: 2394-7659
IMPACT FACTOR- 2.785

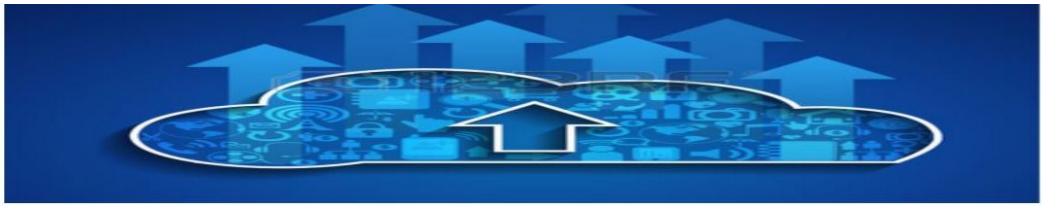
IJERMS

International Journal of Engineering Researches and Management Studies



publish events

File Name c++
File Key 2292
Date 06/01/2016
File Choose File java.txt
submit reset



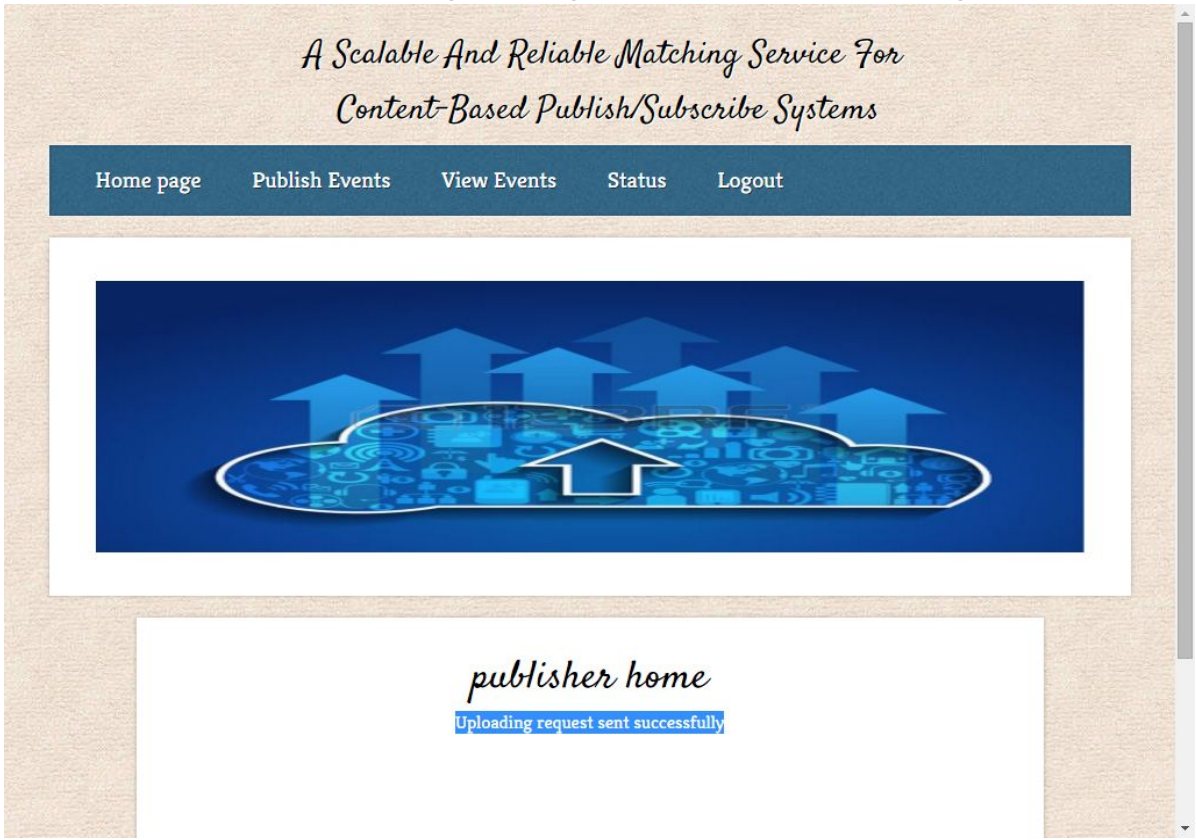
publish events

File Name c++
File Key 2292
Generate Key
Date 06/01/2016
submit reset

ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies



ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies

published events

Publisher UserName	Published Date	File name	File size	Data Center
swe	04/01/2016	java	16622.0	Datacenter4
sai	04/01/2016	java1	16644.0	Datacenter3
swe	04/01/2016	java2	503.0	Datacenter2
swe	05/01/2016	j2	16641.0	Datacenter2
swe	05/01/2016	j5	2892761.0	Datacenter1
swe	05/01/2016	p1	7706.0	Datacenter1
swe	05/01/2016	c	16640.0	Datacenter2

ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies

VII. CONCLUSION

ISSN: 2394-7659
IMPACT FACTOR- 2.785

IJERMS

International Journal of Engineering Researches and Management Studies

This paper introduces SREM, a scalable and reliable event matching service for content-based pub/sub systems in cloud computing environment. SREM connects the brokers through a distributed overlay SkipCloud, which ensures reliable connectivity among brokers through its multi-level clusters and brings a low routing latency through a prefix routing algorithm. Through a hybrid multi-dimensional space partitioning technique, SREM reaches scalable and balanced clustering of high dimensional skewed subscriptions, and each event is allowed to be matched on any of its candidate servers. Extensive experiments with real deployment based on a CloudStack test bed are conducted, producing results which demonstrate that SREM is effective and practical, and also presents good workload balance, scalability and reliability under various parameter settings.

REFERENCE

1. *Xingkong Ma, Student Member, IEEE, Yijie Wang, Member, IEEE, and Xiaoqiang Pei, "A Scalable and Reliable Matching Service for Content-Based Publish/Subscribe Systems" IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 1, JANUARY-MARCH 2015.*
2. *Dataperminite. [Online]. Available: http://www.domo.com/ blog/2012/06/how-much-data-is-created-every-minute/*
3. *F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing user behavior in online social networks," in Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. ACM, 2009, pp. 49–62.*
4. *A. Carzaniga, "Architectures for an event notification service scalable to wide-area networks," Ph.D. dissertation, POLITECNICO DI MILANO, 1998.*
5. *P. Eugster and J. Stephen, "Universal cross-cloud communication," IEEE Transactions on Cloud Computing, 2014.*
6. *R. S. Kazemzadeh and H.-A. Jacobsen, "Reliable and highly available distributed publish/subscribe service," in Reliable Distributed Systems, 2009. SRDS'09. 28th IEEE International Symposium on. IEEE, 2009, pp. 41–50.*
7. *Y. Zhao and J. Wu, "Building a reliable and high-performance content-based publish/subscribe system," J. Parallel Distrib. Comput., vol. 73, no. 4, pp. 371–382, 2013.*
8. *F. Cao and J. P. Singh, "Efficient event routing in content-based publish/subscribe service network," in INFOCOM, 2004, pp. 929–940.*
9. *S. Voulgaris, E. Riviere, A. Kermarrec, M. Van Steen et al., "Sub-2-sub: Self-organizing content-based publish and subscribe for dynamic and large scale collaborative networks," in Research Report RR5772. INRIA, Rennes, France, 2005.*
10. *I. Aekaterinidis and P. Triantafillou, "Pastrystrings: A comprehensive content-based publish/subscribe dht network," in ICDCS, 2006, pp. 23–32.*
11. *R. Ranjan, L. Chan, A. Harwood, S. Karunasekera, and R. Buyya, "Decentralised resource discovery service for large scale federated grids," in e-Science and Grid Computing, IEEE International Conference on. IEEE, 2007, pp. 379–387.*
12. *A. Gupta, O. D. Sahin, D. Agrawal, and A. El Abbadi, "Meghdoot: Content-based publish/subscribe over p2p networks," in Middleware, 2004, pp. 254–273*