

Characterization of a Workload Generator for Content-based Publish-subscribe Research Evaluation

Type: Research Study

Received: January 07, 2026

Published: March 02, 2026

Citation:

Mohamed Bobo Diallo, et al.
"Characterization of a Workload Generator for Content-based Publish-subscribe Research Evaluation". PriMera Scientific Engineering 8.3 (2026): 17-26.

Copyright:

© 2026 Mohamed Bobo Diallo, et al. This is an open-access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mohamed Bobo Diallo^{1*} and Vasilis Sourlas²

¹Universite Felix Houphouet-Boigny, Cote d'Ivoire

²Institute of Communication and Computer Systems (ICCS), Greece

***Corresponding Author:** Mohamed Bobo Diallo, Universite Felix Houphouet-Boigny, Cote d'Ivoire.

Abstract

Achieving large-scale content-based publish-subscribe has been an ambitious research agenda that has received tremendous attention over the last two decades by researchers from distributed systems and networking communities. Simulations have been the most common approach for evaluating solutions. However, the research community on this topic neither has shared workload assumptions nor standard workload generation methodologies. As a result, each effort has introduced its own assumptions and ad-hoc workload generation methodologies. Also, comparison to related alternatives has often been neglected. This has made it difficult to understand the performance gains of one contribution over related alternatives. This paper reports an effort to enhance a workload generation tool for content-based publish-subscribe research using Google groups data. It is enhanced with a visual characterization of the generated workload, given a set of parameters. The workload generated can be characterized in terms of popularity and locality. The resulting software contributes to generating well-specified workloads, facilitates experiment reproducibility, and it will also be time-saving in evaluation processes.

Keywords: performance evaluation; distributed systems; workload characterization; workload generation; content-based publish subscribe

Introduction

Publish/subscribe is one of the fundamental communication style in computer science. It involves subscribing to an interest and being notified with relevant content matching this interest [1, 2].

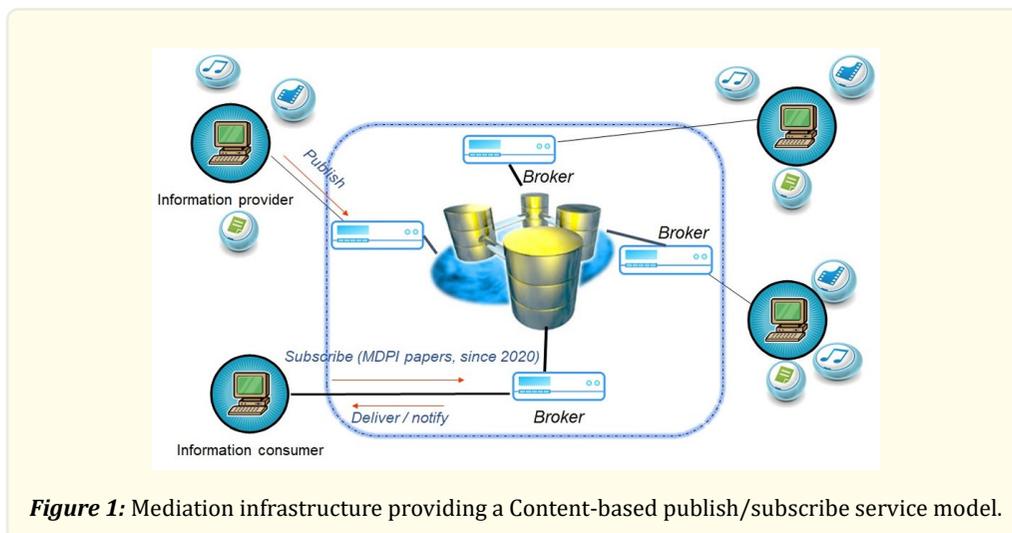
It has three main variants classified according to the expressivity of the subscription model: *channel-based publish/subscribe*, *topic-based publish/subscribe* and *content-based publish/subscribe*.

Channel-based publish/subscribe has been implemented by the RSS and ATOM standards which has enabled subscribing to a feed associated to a web page and to receive related notifications. This has been known as *web syndication* [3].

Topic-based publish/subscribe is a popular pattern that has been implemented in successful distributed software solutions such as Apache Kafka. Internet-based solutions such as Reddit implement this pattern.

Content-based publish and subscribe (CBPS) has been the most expressive variant of this communication style. It supports expressive subscription languages based on attributes/values or keywords. Achieving large-scale CBPS supported by a large infrastructure of brokers geographically dispersed (Fig. 1) has received tremendous attention from many research communities including distributed event-based systems and networking.

It has been an ambitious research agenda, which requires designing efficient solutions and evaluating them often by simulation. Many significant papers have been published on the subject [4-7]. One of the key challenges for evaluating these solutions have been to generate synthetic workload scenarios due to the lack of widely accepted datasets adequate to model realistic workload assumptions. Commonly, each work has introduced its own workload assumptions and workload generation methodology due to the lack of a consensus benchmark or a shared evaluation framework to evaluate solutions. Also, experience has shown that this task of careful workload modeling and generation can be tedious.



One of the early work in assisting the community with a tool for workload generation will be named hereafter YuGen [8]. YuGen interpolates data from Google groups (a topic-based publish-subscribe system) to generate CBPS workloads. However, the properties of the generated workload are not well known from the files generated by YuGen. The most significant workload assumptions impacting the performance of solutions are the popularity of publications and the locality of interests in the topology.

This paper reports our effort to characterize the scenario files generated by YuGen. First, it discusses the main challenges and performance issues in large-scale CBPS research (Section 2). Second, it justifies the importance of cautious workload modeling and characterization for significant performance evaluation (Section 3). Next, we synthesize significant workload assumptions and methodologies reported in previous work (Section 4). Then, we give an overview of YuGen (Section 5). The last section presents the results of the characterization of the distributions generated by YuGen and discusses the impact of the key parameters of the model on those distributions (Section 6).

Large-scale CBPS challenges and Performance issues

Large-scale CBPS is typically characterized by a large number of subscriptions, a high frequency of publications to distribute, and a large supporting infrastructure. The infrastructure contributes to distributing efficiently and timely publications to geographically

dispersed subscribers.

There have been two main functions that this infrastructure should implement. The first is determining the set of subscriptions interested in each publication (*matching* function). The second is efficiently disseminating publications towards geographically dispersed interested subscribers (*dissemination* function). There have been two main approaches to this vision.

The first approach considers *matching* and *dissemination* as independent problems. Some research has considered how to implement a high-throughput matching engine over a cluster of servers [9] or over a rendezvous-based configuration of distributed servers [5, 10]. Other work has focused on the use of *multicast* for disseminating publications to subscribers [11, 12]. The approach consists in grouping similar subscriptions in the same cluster and disseminating each publication through a multicast group set up for the matching subscriptions.

The second approach considers jointly the two functions has been notably implemented in the content-based networking approach [13]. This requires that each content-based router receiving a publication decides where to forward it according to the subscriptions advertised to him. The joint optimization of these two functions has also been considered by Chandramouli et al. [14].

On the importance of workload characterization

Evaluating content-based publish/subscribe solutions (CBPS) requires generating a workload that verifies a set of assumptions that may impact the performance measured (resource consumption, throughput).

A very interesting fact about the importance of workload assumptions on the conclusions one can draw from the evaluation of its solution is the positioning of Riabov et al. in [11] regarding the conclusions on the benefits of multicast in one of the Gryphon¹ papers [15]. Riabov et al., demonstrated that the conclusions from the Gryphon papers were not always true and that they depend on the assumptions made on the locality and similarity properties.

Significant workload assumptions and methodologies

Popularity and *locality* are the most important workload properties worthy to characterize large scale CBPS solutions.

Popularity when it relates to a publication is definable as the number of subscriptions that matches that publication. Some papers have characterized it by the matching distribution (MD) or/and the density of recipients (DR). In the DR case, the popularity also relates to the locality. The DR has also been named *routing accuracy* in some work [16].

Locality refers to the origin of publications and subscriptions and is often tightly related to similarity. Two subscriptions are similar, when they select at least one same publication. Two publications are similar, when they match at least one same subscription. The degree of similarity between two subscriptions (resp. two publications) increases with the set of same publications they select (respectively the set of same subscriptions they match). A common workload assumption is that similar subscriptions are localized in the same region of the network.

Yu et al. discuss workload assumptions from some older contributions [8]. All have not always exhaustively considered the workload patterns relevant to their solution. However, some significant papers have performed comprehensive evaluations. For instance, [5] investigates several different interesting workload assumptions: users interested in events local to them, Zipf and uniform distributions of popularity, most popular events also are the most available. Other work have complementarily tried to interpolate characteristics of existing systems to generate their workloads [12, 7].

We have found three main classes of methodologies for workload generation in reported research on the topic. Two depend on the schema of the target application. This is an original classification. We discuss them below.

Attribute/value schema

First, the workload specification should define the properties of the schema of subscriptions and publications. This may include, attributes types, values domain, number of attributes per publications, number of constraints per subscriptions, relational operators and logical operators supported by the schema. In order to enable a random generation of subscriptions and publications, the characteristics corresponding to each of these parameters should be defined. Such a workload specification can be easily used to generate random subscriptions and publications. Controlling the popularity is a more serious issue, which can be achieved by making some simplifications [17].

Recently, [18] addresses large-scale content-based publish-subscribe implementation over the Kafka framework. It provides a workload specification (each event has 50 attributes, there are 100,000 consumers who submit a total of 250,000 subscriptions, each subscription contains 5 predicates which define conditions on attributes. The attributes of these subscriptions are uniformly selected from the 50 attributes in events).

Other work [19, 20], provide workload specifications describing common parameters such as number of subscriptions, number of events, or average attribute per event. However they lack a clear description of the methodology used and do not provide any workload characterization.

In the case, where the evaluation of the efficiency of a data structure or the performances of a matching algorithm is at stake, the realism of the subscriptions and publications is very important and an over-simplification of the subscription model could make the evaluation superfluous.

Keyword-based schema

Subscriptions and publications are modeled as arrays of keywords. This might match the model of the target application or be a simplification of the real application schema. The workload specification, besides the popularity and locality requirements, may include a set of parameters characterizing the number of keywords per subscriptions/publications or the size of the dictionary. Like the attribute-value model, one has to trade between realism and the ability to control the popularity/locality properties. It is easier to control popularity by setting the size of a subscription to a single keyword.

[6] used this model with a disjunctive semantics for subscriptions and publications where each keyword select matching publications and the whole query select the union of the matching publications. This approach also works with a more conventional conjunctive semantics.

Event-based modeling

Event-based modeling enables the specification of the workload characterization independently of the underlying schema. It is adapted to event-based applications, which assume that real-world events produce publications and that subscriptions register to publications related to those events. This high-level characterization enables a better control of the workload properties. This high-level characterization is used by the workload generator to produce a scenario defining publications, subscriptions and the matching/similarity relationships between them [7, 21].

Note that it is not obvious to bind the generated scenario to a keyword or attribute/value schema. However, it is sufficient in many cases that the workload generator associate to each generated subscription/ publication, any valid representation of the event.

YuGen workload generator overview

The YuGen workload generator works with data originating from Google groups (Ggroups) and Planetlab [8].

It uses the collected data to generate a workload of subscriptions and events (i.e. publications) for a configuration of brokers and

given a variety of parameters that can be used to tune the characteristics of the workload generated. Events are points in a multidimensional space (*event space*) and subscriptions are orthotopes in the event space.

Description of collected data

Ggroups tags each group using three attributes (topic, geographic region, language) and provides statistical data (the number of members, the average quantity of messages posted per month, and the language). Ggroups also provides a hierarchy of topics and regions.

Ggroups defines hierarchies over topics and regions, but it does not provide a hierarchy on languages. So, Yu et al, introduced their own method to classify languages into categories, according to their origins, and set the language hierarchy.

Description of the generated files

YuGen generates three significant files. The subscription files contain for each created subscription its coordinates in the network space and in the event space. By default, a subscription is modeled as a rectangle in the event space. The broker files contain for each created broker, its geographic region and its coordinates in the network space. The event distribution files contains the probability that an event be generated in each each cell of the event space. Each cell is described by its coordinates in the event space.

YuGen approach key insights

YuGen treat topics and languages as dimensions of the event space. Each interest is a pair of topic and language. For each interest, Yu et al. collected a statistical summary for each interest containing the number of messages posted per month in groups associated to that interest and the number of members in groups associated with that interest. Each interest will be associated with a number of subscriptions.

Then, they divide all groups associated with the same interest by their geographic regions and count the number of members within each geographic region. This gives a rough indication of the distribution of subscribers over the network. They then use Planetlab data to obtain precise *network location (coordinates)* for subscriptions. Euclidian distance in the network space between two points help in approximating the network latency [8].

Tuning YuGen workloads

YuGen gives some control to users to customize the generated workloads. For instance, *hot-interest removal* is an option to remove the most popular interests. *Interest diffusion* is another option to smooth the distribution of interests over the event space. Also, *kernel functions* can be used to smooth the distributions of events to comply with a given distribution. All contribute to shape the distribution of interests and events. In determining the subscription count for each interest, YuGen uses a method called *interest generalization*.

YuGen limitations

The most missing feature to YuGen is that it does not help with the simulation of event coordinates in the network space. Moreover, the popularity and locality distribution of the generated workload is not known. Also, the impact of the parameters of the model on the workload properties has not also been studied. This is what has been considered by the eYugen extension described in the next section.

eYugen: Extension of the Yugen workoad generator

eYugen extends YuGen with visual characterization of the workload generated. Users can define the workload size in terms of number of events, subscriptions and brokers. There are additional parameters impacting the workload distributions shape. Some important workload input parameters specific to YuGen are described with options and default values (Table 1). The figures described in this paper, have been obtained with a configuration of 2000 events, 10000 subscriptions and 90 brokers.

eYugen generates two main distributions. A screenshot of a sample simulation is depicted by Fig. 3. The matching distribution is the CCDF² of the event popularity distribution. The popularity of an event relates to the number of matching subscriptions. The density of recipients is the CCDF of the topological popularity of events. As subscriptions are registered to brokers, events are usually disseminated to a set of brokers who have advertised registered subscriptions. The topological popularity of an event is the number of brokers who should receive the event on behalf of their registered subscribers. In the sample distribution depicted in Fig. 3, one can read on the matching distribution that no event has a popularity bigger than 5 % and that there is no event that should be disseminated towards more than 10 % of brokers.

Event space dimension	By default 2, but can be generalized with relevant data
Network space dimension	By default 2, but can be generalized with relevant data
Kernel functions ^a	Uniform, Triangular, Gaussian,
(Kernel function) bandwidth	Epanechnikov positive integer
Interest generalization (^b) mode	None, Uniform, Proportional
Interest generalization hierarchy fraction	floating number ([0, 1])
Interest diffusion (^c) smoothing probability	floating number ([0, 1])
^a Smoothing technique used to generate different distributions according to their shape ^b It is related to the hierarchy of interests. Some subscriptions are related to broad interests rather than specific interests. Interest generalization is the technique for distributing the objective of subscriptions number over the hierarchy of interests. ^c Technique to reduce the variance among similar interests. Implemented to improve the consistency of the workload model	

Table 1: Selected parameters of the YuGen workload generator.

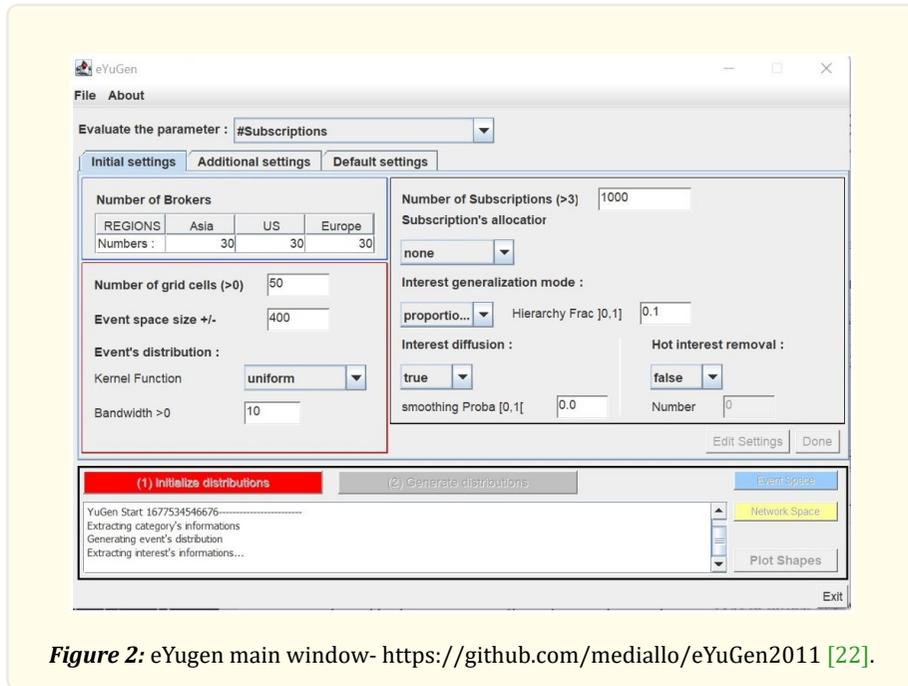


Figure 2: eYugen main window- <https://github.com/mediallo/eYuGen2011> [22].

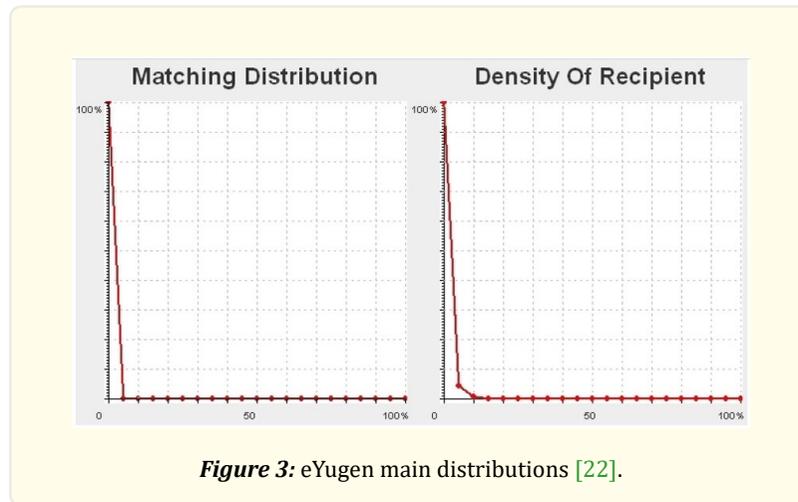


Figure 3: eYugen main distributions [22].

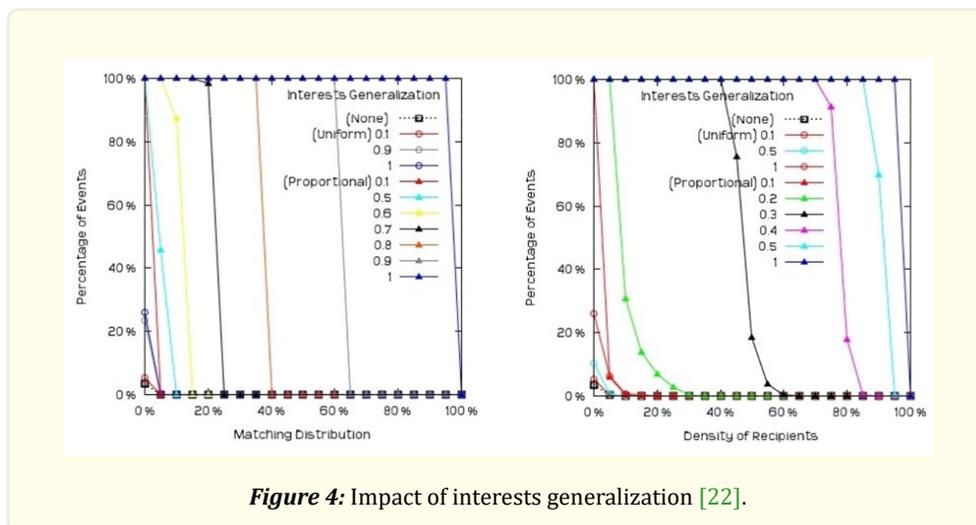


Figure 4: Impact of interests generalization [22].

Discussion of the impact of interest generalization and diffusion on the workload shape

The workload shapes can be customized using various parameters of the model. We report the experiments done for two of the sensitive parameters on the workload generated that are *interests generalization* (Fig. 4) and *interests diffusion* (Fig. 5).

Considering interests generalization, with the *uniform approach*, we observe that there is a large fraction of events that do not match any subscription at all and the popularity of events do not surpass 5% for the generated workload, while the *proportional* approach for the same setting have almost 100 % of events matching at least a subscription and increasing values of the hierarchy fraction parameter [8] increases the maximum popularity. The trend for the density of recipients is similar.

As for interest diffusion the workload while changing is less sensitive to the smoothing probability as compared to the interest generalization mode and hierarchy fraction parameter.

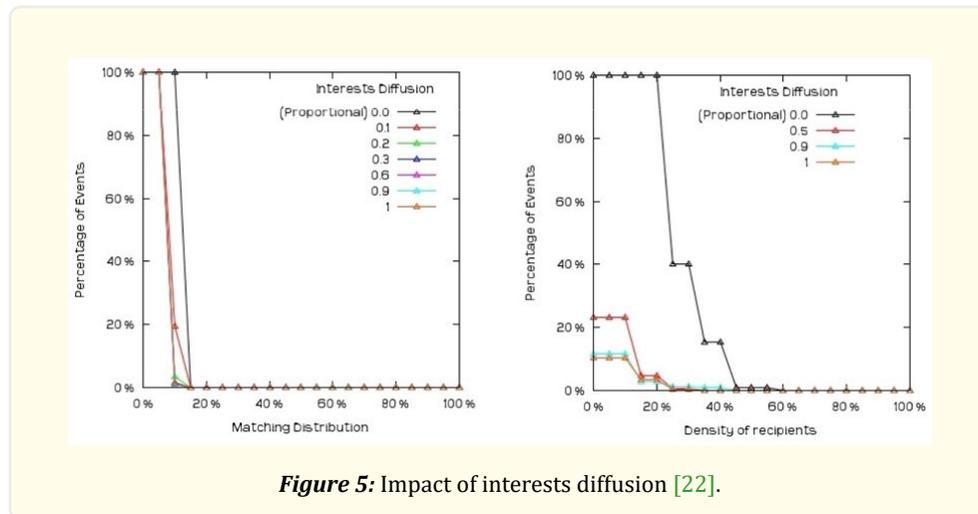


Figure 5: Impact of interests diffusion [22].

Related work

An interesting related work is BEGen a workload generator developed for the BE-Tree project [23]. The originality of BEGen is that it can generate a workload of boolean expressions that matches by construction a given matching popularity³. They also report using DBLP data to generate workloads corresponding to a given matching probability. BEGen is also used for workload generation and work comparison in [16]. It is rich in parameters and can generate more realistic subscriptions and events. However, it is not as easy to use as eYugen and YuGen. Note also that BEGen is posterior to YuGen and is of the synchronicity of the eYugen.

Other early work had called for developing benchmarks for content-based publish/subscribe research [24]. Though being an important and seminal contribution, the work has not been much referenced, but it has inspired a few later work including [25]. More recently a benchmark for content-based publish/subscribe have been proposed [26], but the specification targets social games.

Also, beyond CBPS there are existing benchmark for middleware supporting the publish/subscribe paradigm [27, 28]. In particular, the OpenMessaging⁴ benchmarking suites are available for many messaging systems including Kafka and RabbitMQ.

Conclusion

This paper reports a contribution to the issue of workload generation for evaluating the performance of large scale content-based publish/subscribe research by simulation.

The *eYugen* tool developed enhances a previous contribution with a graphical interface allowing to characterize the generated workload in terms of *popularity* and *locality*, and allows to tailor the generated distributions according to various parameters introduced by the workload generation model described by [8].

This tool can be improved to be used as a reference tool for evaluating contributions by simulations. Each researcher can use it to specify his workload scenario in terms of high-level properties and generate consistent scenario files that can be used for simulations. This will contribute to the reproducibility of experiments and thus improve research quality.

The research community in large-scale distributed systems needs more similar efforts towards a consistent consolidation of the scientific knowledge published on the topic over the last two decades. To the best of our knowledge, each work has introduced ad-hoc workload assumptions and workload generation methodologies making difficult to share a consensual perspective of the advancement of the field with respect to performance issues.

Despite more than two decades of significant research on the topic, Google alerts is currently the only successful content-based publish-subscribe service deployed at global scale. The service has a great potential in this information overload era. Yet, it has obvious relevance problems.

It would be wasteful that so much research effort on the topic fail to concretize the vision of distributed and large scale content-based publish-subscribe possibly decentralized and federated. We believe this topic is of high importance for our time and should not be classified *cold* that early.

Funding

Part of this research has been funded by the Onelab2 FP7 european research project and has been documented in previously released project deliverables [29, 30]. The remaining contributions of the paper have been funded by Université Félix Houphouët-Boigny (Côte d'Ivoire), and not only those above mentioned, from the almighty's benevolence.

Acknowledgments

We are grateful to Mohamed Barkallah [22] for its essential contribution to this work, to Professor Serge Fdida for his mentoring and feedback, to Nicolas Bataille and Fengyu Zhang which has contributed significantly to this conversation. We also thank Dr Albert Yu who has accepted to share its implementation with us. Finally, we would like to thank Dr Akeke Eric Dago for its helpful synchronization to this effort.

References

1. Tarkoma S. "Publish/subscribe systems: design and principles". John Wiley & Sons (2012).
2. Eugster PT, et al. "The many faces of publish/subscribe". ACM computing surveys (CSUR) 35 (2003): 114-131.
3. Ramasubramanian V, Peterson R and Siler EG. "Corona: A High Performance Publish-Subscribe System for the World Wide Web". In Proceedings of the NSDI 6 (2006): 2-2.
4. Carzaniga A and Wolf AL. "Content-based networking: A new communication infrastructure". In Proceedings of the Workshop on Infrastructure for Mobile and Wireless Systems. Springer (2001): 59-68.
5. Cao F and Singh JP. "Efficient event routing in content-based publish-subscribe service networks". In Proceedings of the INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE 2 (2004): 929-940.
6. Majumder A., et al. "Scalable content-based routing in pub/sub systems". In Proceedings of the INFOCOM 2009, IEEE. IEEE (2009): 567-575.
7. Diallo M., et al. "A content-based publish/subscribe framework for large-scale content delivery". Computer Networks 57 (2013): 924-943.
8. Yu A, Agarwal PK and Yang J. "Generating wide-area content-based publish/subscribe work-loads". Network Meets Database (NetDB) (2009).
9. Fabret F, et al. "Filtering algorithms and implementation for very fast publish/subscribe systems". In Proceedings of the ACM SIGMOD Record. ACM 30 (2001): 115-126.
10. Raiciu C, Rosenblum DS and Handley M. "Revisiting content-based publish/subscribe". In Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06). IEEE (2006): 19-19.
11. Riabov A., et al. "Clustering algorithms for content-based publication-subscription systems". In Proceedings of the Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. IEEE (2002): 133-142.
12. Riabov A., et al. "New algorithms for content-based publication-subscription systems". In Proceedings of the Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on. IEEE (2003): 678-686.
13. Carzaniga A, Rutherford MJ and Wolf AL. "A routing scheme for content-based networking". In Proceedings of the INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE 2 (2004): 918-928.
14. Chandramouli B, et al. "ProSem: Scalable wide-area publish/subscribe". In Proceedings of the Proceedings of the 2008 ACM SIG-

- MOD international conference on Management of data. ACM (2008): 1315-1318.
15. Opyrchal L., et al. "Exploiting IP multicast in content-based publish-subscribe systems". In Proceedings of the IFIP/ACM International Conference on Distributed systems platforms. Springer-Verlag New York, Inc (2000): 185-207.
 16. Ji S. "Efficient Content-based Publish/Subscribe Routing and Boolean Expression Matching Algorithms". PhD thesis, Technical University of Munich (2018).
 17. Gupta A., et al. "Meghdoot: content-based publish/subscribe over P2P networks". In Proceedings of the Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware. Springer-Verlag New York, Inc (2004): 254-273.
 18. Qian S., et al. "Fat Topic: Improving Latency in Content-Based Publish/Subscribe Systems on Apache Kafka". In Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications. Springer (2021): 547-558.
 19. Shah M and Kulkarni D. "Design and development of high performance, scalable content based publish subscribe system: doctoral symposium". In Proceedings of the Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems (2016): 406-409.
 20. Qian S., et al. "REIN: A fast event matching approach for content-based publish/subscribe systems". In Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications (2014): 2058-2066.
 21. Diallo MB. "Content-based networking for global scale mediation services". PhD thesis, UPMC Sorbonne Universities (2013).
 22. Barkallah M. "Characterizing a Workload Generator for CBPS systems evaluation". Master's thesis, UPMC Sorbonne Universities (2011).
 23. Sadoghi M and Jacobsen HA. "Be-tree: an index structure to efficiently match boolean expressions over high-dimensional discrete space". In Proceedings of the Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (2011): 637-648.
 24. Carzaniga A and Wolf AL. "A benchmark suite for distributed publish/subscribe systems". Technical report, Colorado Univ at Boulder Dept of Computer Science (2002).
 25. Henjes R, Menth M and Zepfel C. "Throughput performance of java messaging services using sun java system message queue". In Proceedings of the Proceedings 20th European Conference on Modelling and Simulation, Bonn, Germany (2006): 684-691.
 26. Zhang K., et al. "PSBench: a benchmark for content-and topic-based publish/subscribe systems". In Proceedings of the Posters & Demos Session (2014): 17-18.
 27. Lazidis A., et al. "Open-Source Publish-Subscribe Systems: A Comparative Study". In Proceedings of the International Conference on Advanced Information Networking and Applications. Springer (2022): 105-115.
 28. Sachs K., et al. "Performance evaluation of message-oriented middleware using the SPECjms2007 benchmark". Performance Evaluation 66 (2009): 410-434.
 29. Diallo M., et al. "Onelab2 Deliverable: D7.6 - Implementation of content-specific operational extensions". Technical report (2010).
 30. Reason C., et al. "Onelab2 Deliverable: D7.4 - Definition of virtual and federated publish-subscribe architecture". Technical report (2009).

Foot Notes

1. Oneofthepioneering work on content-based publish-subscribe.
2. Complementary cumulative distribution function. It is more practical to highlight the maximum popularity and the absence of interest.
3. <http://msrg.org/datasets/BEGen>.
4. <https://openmessaging.cloud/docs/benchmarks/>.