Final Project: Dynamic Topic Modeling of Systems Research Papers

Theano Stavrinos
SNS Lab
theanos@cs.princeton.edu

Christopher Hodsdon
SNS Lab
chodsdon@cs.princeton.edu

Abstract

This project presents a dynamic topic analysis of top systems papers published at the Symposium on Operating Systems Principles (SOSP) and the Symposium on Operating Systems Design and Implementation (OSDI) over the last five decades. We use dynamic topic analysis [2] to examine the set of topics present in systems papers and how the topics have changed over time. Our results showed that the use of dynamic topic modeling uncovered topics that can be roughly interpreted as systems research topics, though there is some mismatch between the topics and our understandings of the papers. We also found that, as expected, there is a dynamic mix of words in each topic over time.

1 Introduction

Systems conferences are venues for disseminating cutting-edge research on distributed computing systems and related topics. As such, the papers published at these conferences can give insight into the trajectory of computing systems more generally. In this project, we do a longitudinal topic analysis of the papers in the Symposium on Operating Systems Princicples (SOSP) and the Symposium on Operating Systems Design and Implementation (OSDI). These are top-tier systems research conferences that have been held bi-annually since the mid-1960s (SOSP) and mid-1990s (OSDI). The project was inspired by Blei's dynamic topic study of *Science* articles [1, 2].

The motivations for this project are twofold: first, we believe the evolution of computing systems over the last 50 years is of historical interest, and uncovering the trends in systems research illustrates at least part of that trajectory. This includes both differences over time of which topics are included in the conferences, and uncovering shifts over time within the topics themselves. Second, topic analysis is useful in improving the searchability and categorization of these papers. Though the papers are categorized into sessions within each conference (e.g., security, file systems, verification, reliability) and each paper includes a list of relevant keywords, these are only broadly descriptive of the paper's contents. Topic analysis can give a much more detailed representation of what each paper is about.

We use dynamic topic modeling [2] to discover and analyse the topics of SOSP and OSDI publications published in 1991 and later. Our results showed that the use of dynamic topic modeling uncovered topics that can be roughly interpreted as topics we expected, though there is some mismatch between the topics and our understandings of the papers. We also found that, as expected, there is a dynamic mix of words in each topic over time.

The rest of the paper is organized as follows. We discuss related work in Section 2, outline the methods used in Section 3, present our results in Section 4, and discuss possible extensions and ways to improve our results in Section 5. Section 6 concludes the paper.

2 Related Work

This project was inspired by Blei's analysis of *Science* publications over time [1]. The analysis used dynamic topic modeling to uncover changes in word prevalence within topics over time. The same analysis was done for UN speeches in a recent blogpost [7]. Though dynamic topic analysis usually considers full documents, as ours does, the UN speech analysis broke each speech into paragraphs and considered each paragraph as its own document. Because a single speech may cover a wide range of topics, this was necessary to avoid muddling the topics. We discuss doing ths in the context of improving paper searchability in Section 5.2.

A similar approach can also be applied to topic variation across the spatial dimension, rather than the temporal. One study used cascading topic models to examine regional linguistic variation on geotagged Twitter posts [5]. Cascading topic models, like the dynamic topic models used in this project, feed topics learned from one analysis into subsequent models to track topic evolution across a particular dimension. In this case, "pure" latent topics are discovered from the text, and regional variations are discovered with these pure topics as the priors.

Finally, dynamic topic modeling as proposed by Blei and Lafferty [2] makes a Markov assumption about the progression of time. However, this is not the only way to deal with the time component of the analysis. For example, Wang et al. [10] use continuous time rather than discrete time, so that time is a continuous variable in the model.

3 Methods

3.1 Dataset

The full dataset consists of 900 papers published between 1969 and 2018 that were published in the SOSP and OSDI conferences. The papers were collected in PDF format from the ACM Digital Library and individual conference websites using the dblp computer science bibliography to find a listing of paper links [9].

The PDFs were converted to text using the `pdftotext` command [6]. Because of the poor quality of the PDFs in earlier years, we only considered documents from 1991 and later, leaving 663 documents for analysis. We represented each paper as a bag of words with standard preprocessing of the text: common stopwords were removed and words were reduced to their stems. Words that occurred fewer than five times were removed; words that occurred in more than 95% of documents were also removed. After preprocessing, the dictionary consisted of 12,743 words, of which we used the top 1000 most frequent words to reduce the processing time.

3.2 Dynamic Topic Modeling

This section describes Blei and Lafferty's dynamic topic model and how it was used to discover latent topics throughout the last three decades of SOSP and OSDI papers [2].

3.2.1 How it works

The dynamic topic model views each timepoint as an instance of a Latent Dirichlet Allocation (LDA) model. LDA views each document as a bag of words generated from a topic distribution. Specifically, a document's topic proportions are generated from a Dirichlet distribution parameterized by α. Those topic proportions are used to generate a word position's topic assignment, z. The word for that position, the observed parameter w, is generated from another Dirichlet-distribution-generated value, β. This represents the probabilities of each word in the vocabulary being selected for a given topic. The β distribution is in turn parameterized by η (not shown in Figure 1).

The dynamic topic model extends the LDA model over time. Each β is dependent on the β in the previous timepoint. This is to reflect changes in topics over time: a topic may change, but it cannot change arbitrarily; it is generally affected by the topic's word distributions in the past. Similarly, α also depends on the α in the previous timepoint.

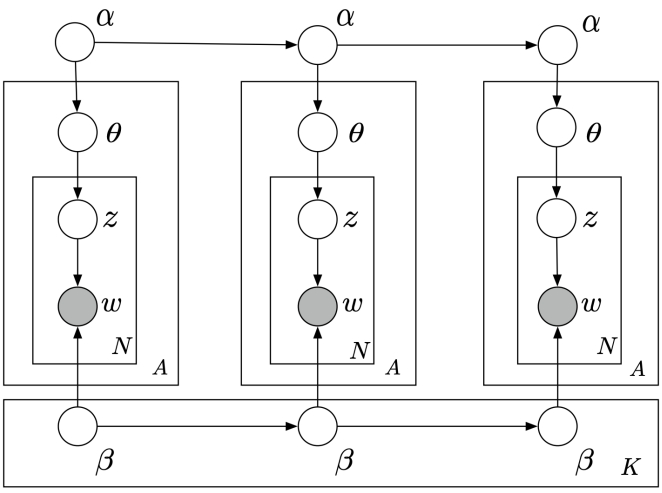

Figure 1: The graphical model for the dynamic topic model. α: Dirichlet distribution parameter to draw a document's topics. θ: per-document topic proportions. z: the word's topic assignment. w: nth word for document d. η: hyperparameter for a Dirichlet distribution generating β, the set of topics and their word proportions. A: the number of documents. K: the number of topics. N: the number of words in a document. Each vertical plate corresponds to a timepoint; prior word distributions per topic (β) and topic distribution parameter (α) for each year in the model depend on the α and β from the previous timepoint.

3.2.2 How we used it

Analysis was done in Python3 using the `gensim` topic modeling library [8]. The model was generated with `gensim`'s `DtmModel` class, which is a wrapper around an optimized C implementation of DTM.

We ran the analysis with 5, 7, and 10 topics (components). We chose these values because conferences typically have 10-12 sessions, a couple of which cover the same topic (e.g., there might be two separate operating systems sessions), meaning there are around 7-9 unique session topics per conference. We found that, with 10 topics, it was difficult to come up with interpretations that didn't overlap. We felt that 5 topics missed some nuance; for example, the topics did not appear to distinguish between "file systems" and "operating systems". For this reason, we conducted the rest of our analysis with 7 topics.

4 Results

The dynamic topic model generated topics which we could generally interpret as systems research subjects. By examining the top 10 words over time for each of the topics, we came up with the following topic labels:

Topic 0: Storage/file systems

Topic 1: Databases

Topic 2: Security/usability

Topic 3: Networking

Topic 4: Distributed databases

Topic 5: Analytics/scheduling

Topic 6: Operating systems

The top 10 most probable words over time for all topics, from which we generated the topic names, are given in the appendix.

These topics correspond roughly with systems topics, a few subjects that are nearly always represented by a separate session at OSDI or SOSP were not definitively suggested by a single model-generated topic. In particular, neither verification nor debugging are purely represented in the learned topics. Topic 6 contains many words that indicate a focus on debugging, but only in later years (see Figure 2). It is not clear why this is.

1991	2005	2018
thread	kernel	execute
kernel	driver	program
user	code	code
processor	memory	thread
schedule	execute	failure
space	function	test
address	thread	bug
block	linux	analysis
execut	check	develop
interrupt	program	event

Figure 2: The top 10 words over time, sorted, for Topic 6 (operating systems). This corresponds to the β parameter in the dynamic topic model (see Figure 1).

We expected a dynamic mix of words in each topic over time, and our results confirm this. Figure 3 shows, for each topic, a selection of words and the evolution of their probability in the topic over time (the β parameter in Figure 1). The figures show that the probabilities of words are not stable over time. Words can be highly probable at one timepoint and highly improbable at others within a topic; see, for example, *kernel* in Topic 6.

Words can also "switch" from one topic to another. An example is the word *file*: it is very probable in Topic 0 (storage/file systems) and Topic 5 (analytics/scheduling) at first, but becomes much less probable in later years. As *file* becomes less probable in these topics, its probability rises dramatically in Topic 1 (databases). What seems to have happened is that *file* began co-occurring with words in Topic 1 and eventually became part of that topic's lexicon, disappearing entirely from the other two topics. However, it is difficult to draw definitive conclusions about why this happened and how this impacted the model more generally.

5 Discussion

We set out to automatically identify topics in systems papers, and how those topics changed over time. The results show that the topics detected by dynamic topic modeling are interpretable as roughly corresponding to actual topics in systems research, listed in Section 4, though a few key topics (namely, debugging and verification) aren't purely represented as topics. This may be because these tend to be techniques applied to types of systems (file systems, networks, operating systems, etc.) and so verification/debugging papers are distributed among the systems' topics rather than categorized as their own topic.

An interesting finding is that the evolution of a single topic can reflect changes in focus for that topic over time. For example, we show in Figure 2 the top 10 most probable words for Topic 6 over time. At earlier timepoints, this topic is strongly identifiable as "operating systems", given the words *thread*, *kernel*, *schedule*, etc. as very probable. As time advances, the top 10 lists begin to include words more closely associated with correctness and debugging, e.g., *check*, *analysis*, *failure*, and *bug*. This is potentially because the focus on operating systems became more about proving correctness and debugging of operating systems using techniques which can be applied to different types of systems, as described above.

5.1 Document spotlight: MapReduce

We were interested in how the topic model assigned topics to actual papers, and how well these topics meshed with our understandings of the papers. For this we examined the MapReduce paper [3]. MapReduce was published in 2004 in OSDI. The key contribution of the work is a system for easily

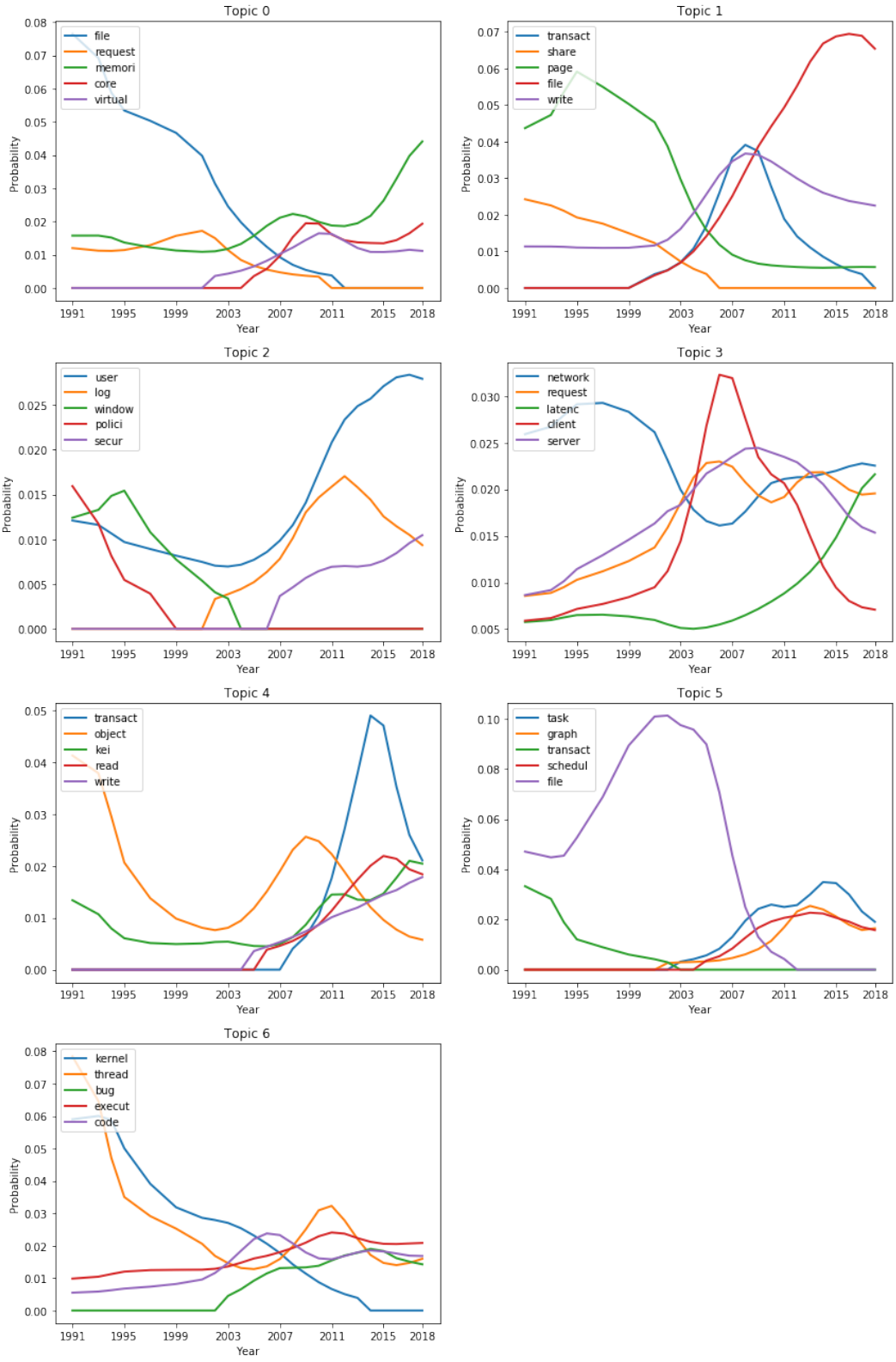

Figure 3: The weightings of words over time for 7 topics (the β parameter).

Topic	Topic proportion
0 (Storage/file systems)	0.046
1 (databases)	0.008
2 (security/usability)	0.533
3 (networking)	0.270
4 (distributed databases)	0.003
5 (analytics/scheduling)	0.003
6 (OS)	0.136

Figure 4: The topic proportions for MapReduce, which is the θ parameter in the dynamic topic model (see Figure 1).

specifying and executing a class of distributed computation on clusters; the system transparently handles the hard work of distributing the computation (e.g., breaking up the data and computation into chunks, handling device failure). The work has been hugely influential in both the systems and the wider computer science community, with the Communications of the ACM version of the paper currently cited over 26,000 times [4].

The topic proportions (the θ parameter) for the MapReduce paper are listed in Figure 4. Because the paper has a strong focus on abstracting a complex system—a cluster—for ease of use by typical programmers, it makes sense that its most probable topic is Topic 2 (security/usability) with a weight of 53%. The next most probable topic is Topic 3 (networking), which is also to be expected since it is a highly-distributed system and it must deal with the performance and practicality consequences of moving data across the network.

MapReduce is a framework for distributed computation, handling the scheduling of tasks within the framework itself and allowing ease of data analytics. Therefore, we did not anticipate the low probability of Topic 5, analytics/scheduling, given that this is the main focus of the paper. However, this topic proportion may make sense for MapReduce. MapReduce is a *framework* for data analytics, this means it may not discuss distributed data analytics much beyond how the framework can be used for data analytics. This means that MapReduce discusses analytics in a very different way than papers whose main focus is distributed data analytics or scheduling! This subversion of expectations can allow us to more deeply understand the actual topics that papers are concerned with and can lead to more interesting questions: can we use MapReduce (or a similar framework) for problems beyond analytics?

5.2 Extension: search

One of our goals with this work was to find more detailed paper categorizations than the provided session topics or paper keywords. This would make it easier to find papers that contain a particular topic, but for which the topic is not the primary focus of the paper.

Searching the papers via topics can also give us a better understanding of how topics have changed over time. The topic word probabilities suggest how the topics have evolved, for example the word "transaction" greatly increasingly its probability after 2017 for Topic 4 (distributed databases). Seeing the change in the use of words gives us some insight as the ideas represented in the topic have changed to be more about transactions. Being able to search through papers over time by topic allows deeper understanding than words becoming more prevalent.

For the purposes of searchability, it may also be useful to do a more granular topic analysis, on paper sections. This would give us the extra dimension of being able to see which sections are likely to be about which topics. This would allow us to to look at subsections of the papers to see the most related sections to the topic in question.

The above could be accomplished by using the topic proportion assignment given to each document in OSDI and SOSP (θ). A query for documents in a topic could consist of the parameters: `topic`, `min_topic_proportion`, `time_slices`, and `n_docs`. The semantics of the query could be such that for each time slice in `time_slices` we return `n_docs` in descending order of topic proportion for `topic`, given it is higher than `min_topic_proportion`. We could extend this query by also returning the location of "most dense" discussion in the document for that paper. That

is, we would find clusters of words that have a high probability for that topic during that time slice. This search query would enable us to browse documents via topics throughout time. This could give us a richer view of topics and their evolution in time.

5.3 Potential improvements

This work could be improved in several ways.

- PDF quality for earlier conference years was poor; oftentimes the PDF was a photocopy of a type-written paper. Because of this we had to leave out earlier years from the topic model. In general the PDF-to-text conversion tool we used, `pdftotext`, seemed to struggle with many of the papers, perhaps due to the two-column formatting and use of figures, tables, and footnotes in the papers. Future work could use a more sophisticated optical character recognition (OCR) tool to convert the papers to text more faithfully. This would enable a more accurate and more complete analysis of the papers over time.
- For the same reason, we did not do n-gram analysis. Many of the text versions of the papers were mangled not just in the identifications of words as such, but also in the orderings of words in relation to one another. A common kind of mis-parsing broke lines into chunks of a few words, and interleaved these chunks in an order that had little relation to their ordering in the paper; we believe this had something to do with the column formatting of the paper. With more sophisticated text parsing, we could use n-grams to capture important systems concepts like "distributed transactions" and "state machine replication".
- We do not include any metadata that is not in the papers themselves (e.g., titles and author names are included, but the session in which the paper appeared at the conference is not included). We also do not treat author names, institutions, and bibliographical references differently from regular text. We believe these are justifiable choices; author names are likely to crop up repeatedly in related papers and their references. It would be interesting to explore whether considering them as separate kinds of datapoints leads to different latent topics and/or document-topic weights.

All of the above factors may contribute to the fact that some papers' topic weights (the θ parameter) do not always correspond to our understandings of the papers. Two additional possibilities are that we are simply mislabeling the topics learned by the model, and that we do not fully understand how topics evolve over time. Identifying the cause(s) of the unintuitive paper topic proportions is an area for future work.

6 Conclusion

In conclusion, we analysed over 600 top-tier research papers using dynamic topic modeling with the goal of understanding topic change over time and of getting a more nuanced categorization of papers than we might from conference session names and paper keywords. We discovered that dynamic topic analysis opens up rich avenues for better understanding systems papers both as a body of documents and as individual works. The topics discovered by the dynamic topic model roughly corresponded to topics we recognize as systems research subjects, though more time needs to be devoted to interpreting the topics and understanding why papers were categorized as they were.

References

[1] David Blei, Lawrence Carin, and David Dunson. Probabilistic topic models: A focus on graphical model design and applications to document and image analysis. *IEEE signal processing magazine*, 27(6):55, 2010.

[2] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[3] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters, 2004. In *OSDI: Symposium on Operating System Design and Implementation*, 2004.

[4] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[5] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1277–1287, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[6] Glyph & Cog. `pdftotext`. `https://linux.die.net/man/1/pdftotext`, 2004. Accessed 1-May-2019.

[7] Luke Lefebure. Exploring the UN General Debates with Dynamic Topic Models. `https://towardsdatascience.com/exploring-the-un-general-debates-with-dynamic-topic-models-72dc0e307696`, 2018. Accessed 1-May-2019.

[8] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[9] Schloss Dagstuhl. dblp computer science bibliography. `https://dblp.uni-trier.de/`, 2019. Accessed 1-May-2019.

[10] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

A Appendix

The following are the top 10 most probable words for each topic (parameter β) over time, along with the interpretation of each topic.

Topic 0: Storage & file systems

1991	2005	2018
file	disk	memory
block	block	page
disk	page	thread
cache	file	kernel
read	memory	core
write	manage	hardware
memory	power	virtual
storage	kernel	alloc
segment	cache	cpu
request	device	usenix

Topic 1: Databases

1991	2005	2018
page	write	file
memory	disk	crash
share	block	write
lock	transact	disk
object	page	log
fault	memory	block
program	file	state
table	commit	node
cache	read	workload
processor	machine	directory

Topic 2: Security/usability

1991	2005	2018
cache	detect	user
policy	analysis	server
program	value	message
miss	problem	trace
window	error	secure
user	attack	inform
instruction	model	path
effect	machine	log
trace	user	observe
interval	state	network

Topic 3: Networking

1991	2005	2018
message	node	network
network	client	latency
node	request	request
link	server	resource
protocol	network	server
receive	service	cache
send	failure	service
communication	protocol	usenix
state	message	load
packet	response	storage

Topic 4: Distributed databases

1991	2005	2018
object	secure	transact
interface	code	key
key	packet	read
user	host	write
layer	user	store
type	object	throughput
code	network	hash
domain	address	value
service	label	client
call	control	server

Topic 5: Analytics/scheduling

1991	2005	2018
file	file	query
log	user	task
server	server	graph
transact	client	schedule
client	group	stream
write	directory	xad
update	record	optimal
directory	read	model
manage	device	usenix
user	log	execute

Topic 6: Operating systems

1991	2005	2018
thread	kernel	execut
kernel	driver	program
user	code	code
processor	memory	thread
schedule	execute	failure
space	function	test
address	thread	bug
block	linux	analysis
execut	check	develop
interrupt	program	event