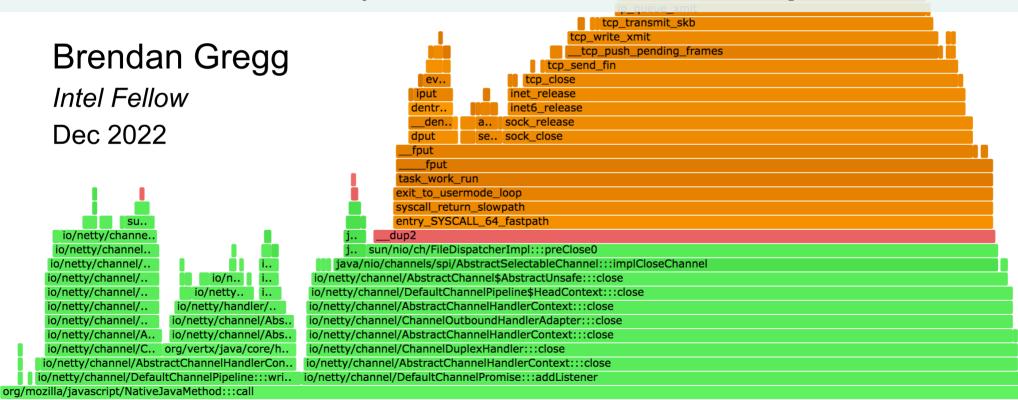
YOW! 2022

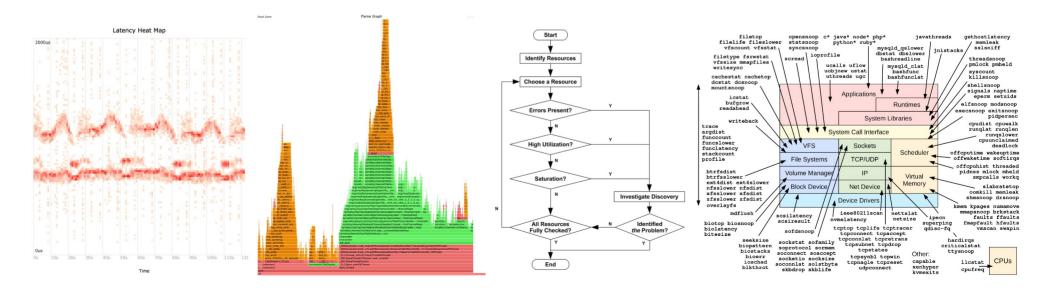
Visualizing Performance The Developer's Guide to Flame Graphs



I'd like to begin by acknowledging the Traditional Owners of this land and pay my respects to Elders past and present.

My Dream

To Completely Understand the Performance of Everything

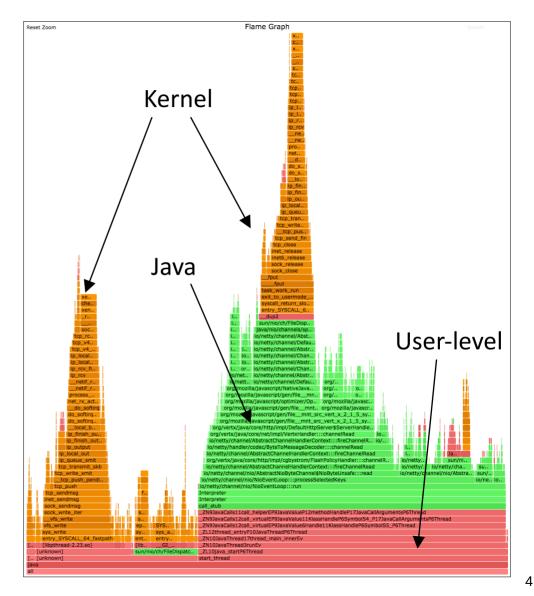


Flame Graphs

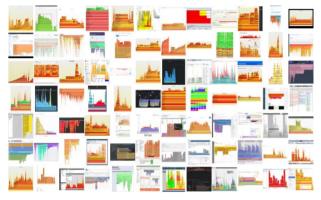
A visualization of software

Can also visualize CPU and other resource usage

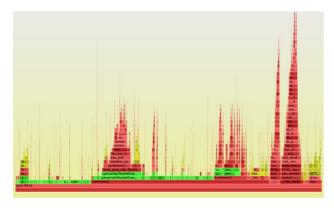
Now a staple in performance engineering



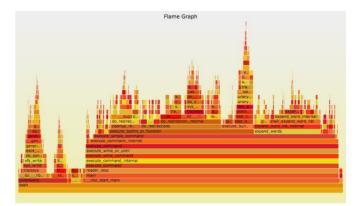
Agenda



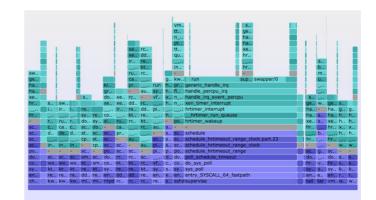
1. Implementations



3. Stacks & Symbols



2. CPU Flame graphs



4. Advanced flame graphs

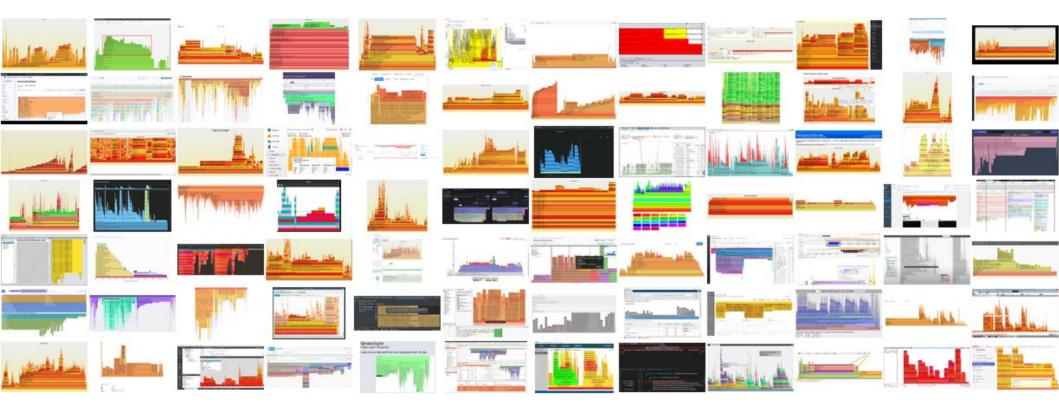
Take Aways

- 1. Interpret CPU flame graphs
- 2. Understand runtime challenges
- 3. Why eBPF for advanced flame graphs
- A new tool to lower your cost, latency, and carbon

Slides online: https://www.brendangregg.com/Slides/YOW2022_flame_graphs.pdf

1. IMPLEMENTATIONS

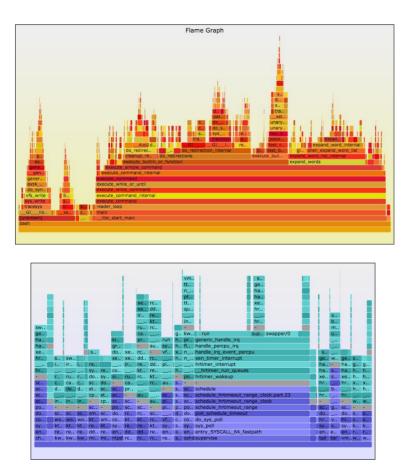
Quick Tour of Some Examples

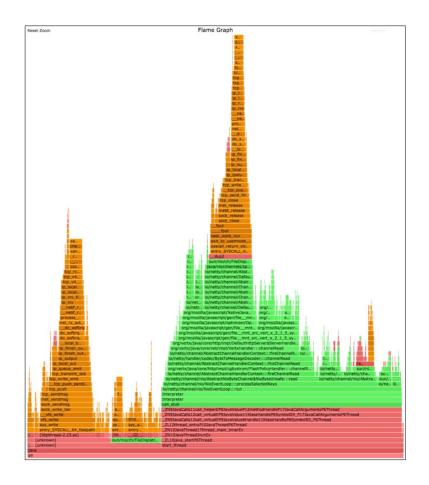


More examples in later "bonus slides" section.

(Note: This is not an an endorsement of any company/product or sponsored by anyone.)

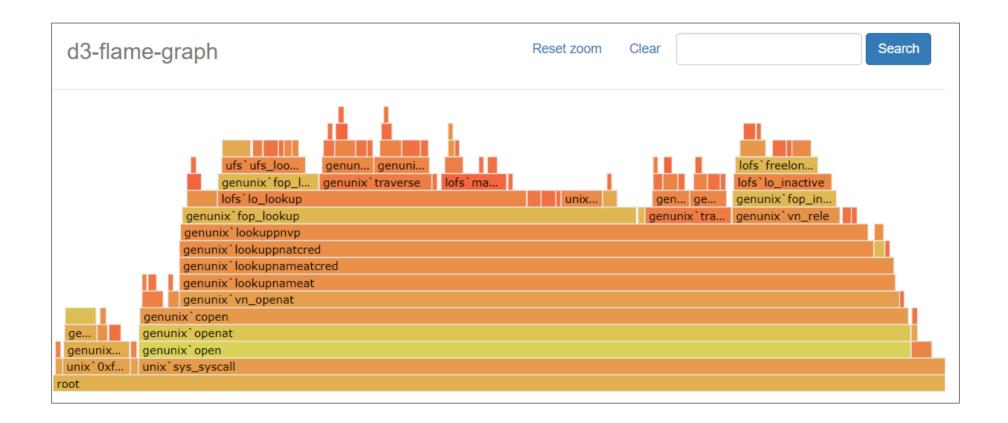
My original flamegraph.pl (2011; using Perl/SVG/JavaScript)





https://github.com/brendangregg/FlameGraph

Martin Spier d3-flame-graph (my colleague at Netflix; 2015; D3)



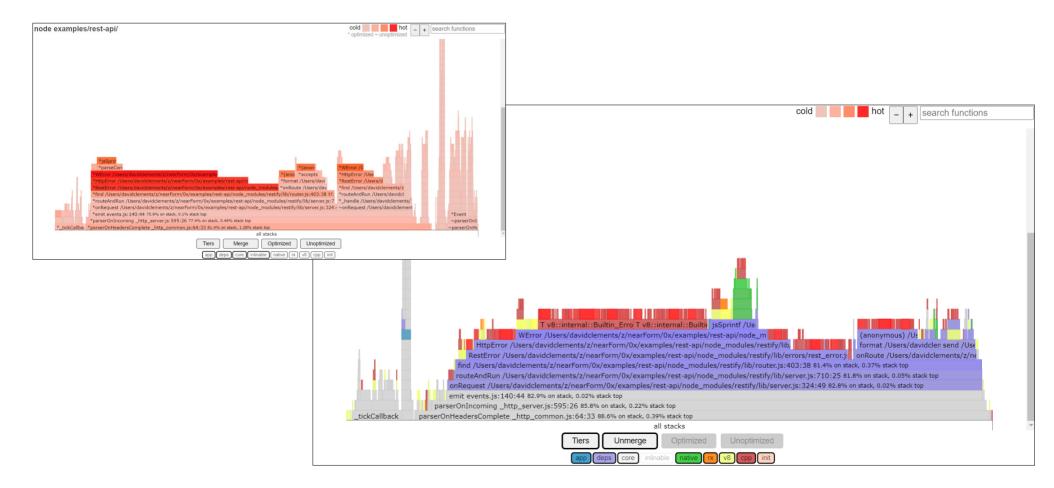
Source: https://github.com/spiermar/d3-flame-graph https://martinspier.io/

Facebook: Strobelight (2014)

ceremer > >: 174565		
dena: 172512		
ot, doead: 172512		l I
ache shrift concurrency. PhreadThread shreadMandvoid*1 157830		facebook concurre
	re-spacks shrift concurrency. ThreadManager. Impl7 cfacebapacke shrift Cpp2Worker.serveD. 36022	facebook concurr
	folly Baton cold atomics a pip. 20017 Feinteen Jandiertfred20dfacebook/smultifeed1 (billy EventRase ScopPorever): 36022	14209 facebook concurr
about multified ranking BoulderStoryScorer weighttoryWithDebugflacebook multifies fo	1 - 4248 Internetionek multifield RankingTack rankifarabook multifi Asily EventRase Joop() 36022 - 44444	facebook concurr and Junction, has
abook multifeed ranking BoulderStoryScorer weightStorySacebook multifeed ranking fo		00. facebook multifier facebook multifier
shock multifield ranking CTRReigher weighttoryffacebook multifield Feedflordacelyfs	A facebook multifield ranking facfacebook multific. His fevent process, active 11022	facebook multifier
abook multifeed ranking CTRItoryPredictor getPredictionstate velocabourlaceto face? (1	a bold alterator charabook multifies 10248 or 60 byd c. Pyfarzbook multified ranking Farfarzbook oftid a 6,21 ispache shrift arans. TAfoliy RothizationQu-	facebook multifier
abook multifield raiking FeaturePipeline processReppfacebook faceborfacebo faile	e bitd alsozator charabook multifercenske: c.d. E.a.O. Labort: 19107 ("dood"), void"s Terferchook multifield ranking Boffarationfa , e i.i.b? departe thirth aryn: TA JohfulyBoenbase	
a afforator characteristic multifield carding COTeatValues and afforated afford all old all the a about multifield carding featuracteristics multifield cardinated factor factor factor	1 E-bit almoster classionsk multifierfaration it) = a ((.h. 13008 13798 1 pfarsbook multifierd ranking Milliof Fig. 6 f = a (r.s.), apartie shrift Cap2Cha-fully Evendbase runs	1.9982 Fac alter fac alter Card
a altocator classicosk multifield std altocator classicosk and altoget at the face of state and face of state face back multifield ran face 1, 5 1 5 5 5 1 1	E-Md affocator charabook multite (Er EF (E) 8 (EF) Kingt 18700 (01-5: 13739) Sectionsk multitead ranking 1: (A) (-5:) (-1:) (Distancheditiviti)1996, (Dist) 7, Function, 9:	Sacabo facebo Grup
a shocator characteristic multifield facebook imultifield. File III 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	E-tat allocator charabook multi- 61 - 61 9 - 61 9 - 81 9 - 8700 - 11045 - 6 Granabook multificatacelo facet 11 - 61 - 1 - gapache, stoch HeaderO facebook multificatio	
a should be the set of	E-Indi alfocator chaostal alfaceb si si si and default, delete chab, and default, delete chab, and default, delete f. (anabook: multifierfacilantfaceb si si), (2NdPacebook: Elsenvice facebook: multifield)	
	E chd. shlocator charaidd cai cong. In: 1. 18700 12222 0041 yr farabolydarfyffill farflara 1. Farabook, sarvicemular farabook, mulo Gold X.	
	E old adulterification (2010) et al. 2010	1
A.M. 1.1.101	E dat Exclosionitant de de cento in la facebullatigatat al 6 (1184) in (2008) antes de cento	
	E eine anterenteten for De State Berten Berten Berten Berten Ber	
	E and annual data and a state of the state o	
	MAND 8 1 1 1/6 00 0 10 Sector/day 1813 00, 7742 3360 (
	martin all all all all all all all all all al	
	And S S S S S S S S S S S S S S S S S S S	
	and a state and a stat	1.1
	1 Sana Sana Sana Sana Sana Sana Sana San	1
	· · · · · · · · · · · · · · · · · · ·	
	Balling of the Same	1

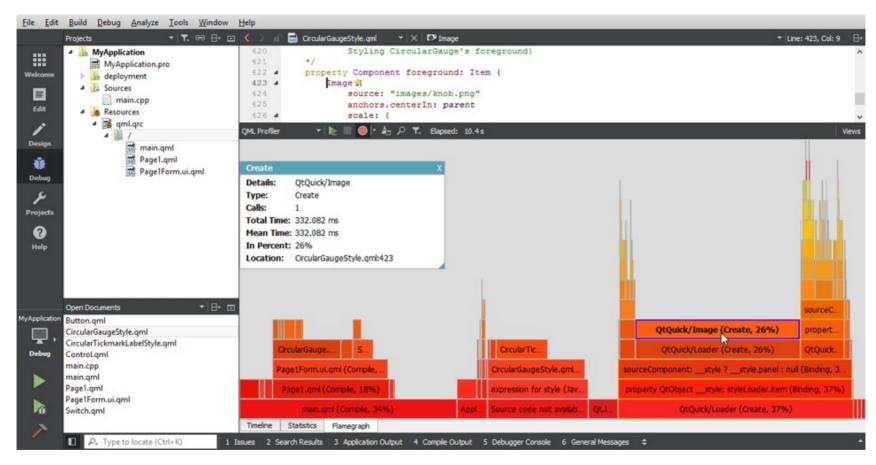
Source: https://tracingsummit.org/ts/2014/files/TracingSummit2014-Tracing-at-Facebook-Scale.pdf

Node.js: 0x (2016)



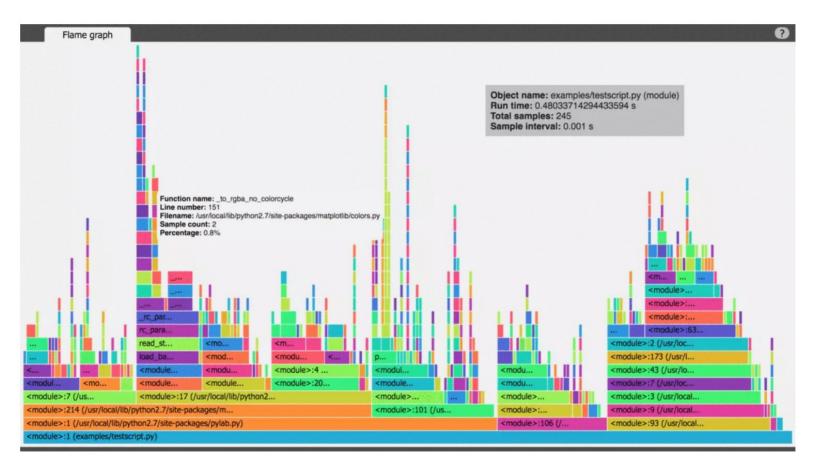
Source: https://github.com/davidmarkclements/0x (David Mark Clements)

Qt: Creator (2016)



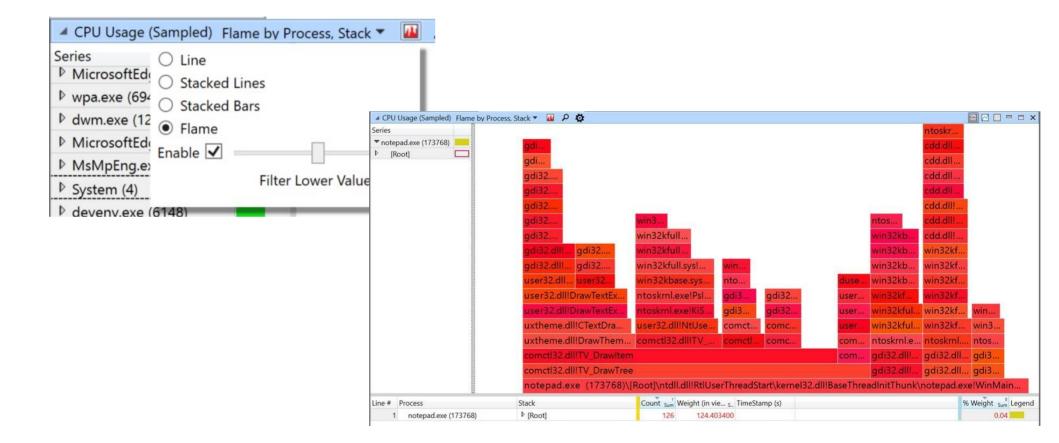
Source: https://www.qt.io/blog/2016/05/11/qt-creator-4-0-0-released

Python: vprof (2016)



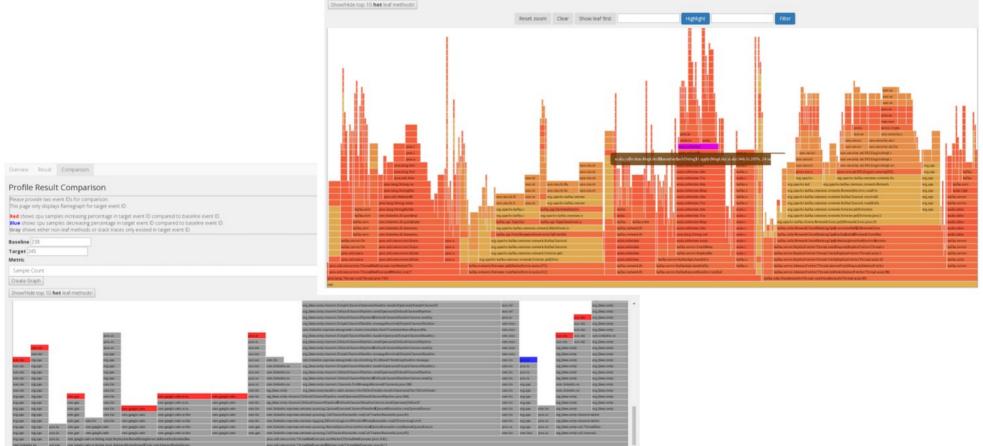
Source: https://github.com/nvdv/vprof (Nick Volynets)

Microsoft: WPA / ETW (2016)



Source: https://learn.microsoft.com/en-us/windows-hardware/test/wpt/graphs#flame_graphs

LinkedIn: ODP (2017)



Source: https://engineering.linkedin.com/blog/2017/01/odp--an-infrastructure-for-on-demand-service-profiling

Oracle: Developer Studio Performance Analyzer (2017)

org. a pache. hadoop org. a pache. b adoop org. a pache. Badoop io. SequenceFile SR org. a pache. hadoop. io. SequenceFile SR org. a pache. hadoop. org. a pache. Badoop. mapreduce. Iib. input. SequenceFileRecordReader. r org. a pache. mahout. clorg. a org. a pache. mahout. clorg. a org. a pache. mahout. clorg. a org. a pache. hadoop. mapreduce. Iib. input. SequenceFileRecordReader. r org. a pache. mahout. clustering org. a pache. hadoop. mapred. MapTask. runNew Mapper(org. a pache. hadoop. mapred. JobConf, org. a pache. hadoop. mapred. MapTask. runNew Mapper(org. a pache. hadoop. mapred. JobConf, org. a pache. hadoop. mapred. TaskUmbilicalProtocol) org. a pache. hadoop. mapred. ChildS4. run() Java Java security. AccessControlContext, java. security. AccessControlContext, java. security. AccessControlContext, java. security. AccessControlContext, java. security. AccessControlContext)	me 🔻	
org.apache.hadoop.mapreduce.MapContext.nextKeyValue() org.apache.mahout.clustering.org.apache.mahout.clusterini.org.a org.apache.hadoop.mapreduce.Mapper.run(org.apache.hadoop.mapreduce.Mapper\$Context) org.apache.hadoop.mapred.MapTask.runNewMapper(org.apache.hadoop.mapred.JobConf, org.apache.hadoop.mapreduce.split.JobSp org.apache.hadoop.mapred.MapTask.run(org.apache.hadoop.mapred.JobConf, org.apache.hadoop.mapred.TaskUmbilicalProtocol) org.apache.hadoop.mapred.Child\$4.run() Java_java_security_AccessController_doPrivileged_Ljava_security_PrivilegedExceptionAction_2Ljava_security_AccessControlContext) java.security.AccessController.doPrivileged(java.security.PrivilegedExceptionAction, java.security.AccessControlContext) java.security.auth.Subject.doAs(java.security.auth.Subject, java.security.PrivilegedExceptionAction) org.apache.hadoop.mapred.Child.main(java.lang.String[])	o o o construction o o construction o o construction o constructio	
org.apache.hadoop.mapreduce.Mapper.run(org.apache.hadoop.mapreduce.Mapper\$Context) org.apache.hadoop.mapred.MapTask.runNewMapper(org.apache.hadoop.mapred.JobConf, org.apache.hadoop.mapreduce.split.JobSp org.apache.hadoop.mapred.MapTask.run(org.apache.hadoop.mapred.JobConf, org.apache.hadoop.mapred.TaskUmbilicalProtocol) org.apache.hadoop.mapred.Child\$4.run() Java_java_security_AccessController_doPrivileged_Ljava_security_PrivilegedExceptionAction_2Ljava_security_AccessControlContext) java.security.AccessController_doPrivileged[java.security.PrivilegedExceptionAction, java.security.AccessControlContext) java.security.auth.Subject.doAs(javax.security.auth.Subject, java.security.PrivilegedExceptionAction) org.apache.hadoop.security.UserGroupInformation.doAs(java.security.PrivilegedExceptionAction) org.apache.hadoop.mapred.Child.main(java.lang.String[])	-	0
org.apache.hadoop.mapred.MapTask.runNewMapper(org.apache.hadoop.mapred.jobConf, org.apache.hadoop.mapreduce.split.jobSp org.apache.hadoop.mapred.MapTask.run(org.apache.hadoop.mapred.jobConf, org.apache.hadoop.mapred.TaskUmbilicalProtocol) org.apache.hadoop.mapred.Child\$4.run() Java_java_security_AccessController_doPrivileged_Ljava_security_PrivilegedExceptionAction_2Ljava_security_AccessControlContext java.security.AccessController_doPrivileged[java.security_PrivilegedExceptionAction_java.security_AccessControlContext) java.security.accessController.doPrivileged(java.security.PrivilegedExceptionAction, java.security.AccessControlContext) javax.security.auth.Subject.doAs(javax.security.auth.Subject, java.security.PrivilegedExceptionAction) org.apache.hadoop.security.UserGroupInformation.doAs(java.security.PrivilegedExceptionAction) org.apache.hadoop.mapred.Child.main(java.lang.String[])	pacne.manout.clust.cc	0
org.apache.hadoop.mapred.MapTask.run(org.apache.hadoop.mapred.JobConf, org.apache.hadoop.mapred.TaskUmbilicalProtocol) org.apache.hadoop.mapred.Child\$4.run() Java_java_security_AccessController_doPrivileged_Ljava_security_PrivilegedExceptionAction_2Ljava_security_AccessControlContext java.security.AccessController.doPrivileged(java.security.PrivilegedExceptionAction, java.security.AccessControlContext) java.security.auth.Subject.doAs(javax.security.auth.Subject, java.security.PrivilegedExceptionAction) org.apache.hadoop.security.UserGroupInformation.doAs(java.security.PrivilegedExceptionAction) org.apache.hadoop.mapred.Child.main(java.lang.String[])	litSTaskSplitIndex.org.au	0
org.apache.hadoop.mapred.Child\$4.run() lava_java_security_AccessController_doPrivileged_Ljava_security_PrivilegedExceptionAction_2Ljava_security_AccessControlContext java.security.AccessController.doPrivileged(java.security.PrivilegedExceptionAction, java.security.AccessControlContext) java.security.auth.Subject.doAs(javax.security.auth.Subject, java.security.PrivilegedExceptionAction) org.apache.hadoop.security.UserGroupInformation.doAs(java.security.PrivilegedExceptionAction) org.apache.hadoop.mapred.Child.main(java.lang.String[])		
ava_java_security_AccessController_doPrivileged_Ljava_security_PrivilegedExceptionAction_2Ljava_security_AccessControlContext ava.security.AccessController.doPrivileged(java.security.PrivilegedExceptionAction, java.security.AccessControlContext) avax.security.auth.Subject.doAs(javax.security.auth.Subject, java.security.PrivilegedExceptionAction) org.apache.hadoop.security.UserGroupInformation.doAs(java.security.PrivilegedExceptionAction) org.apache.hadoop.mapred.Child.main(java.lang.String[])		orc o o
ava.security.AccessController.doPrivileged(java.security.PrivilegedExceptionAction, java.security.AccessControlContext) avax.security.auth.Subject.doAs(javax.security.auth.Subject, java.security.PrivilegedExceptionAction) org.apache.hadoop.security.UserGroupInformation.doAs(java.security.PrivilegedExceptionAction) org.apache.hadoop.mapred.Child.main(java.lang.String[])		
avax.security.auth.Subject.doAs(javax.security.auth.Subject, java.security.PrivilegedExceptionAction) org.apache.hadoop.security.UserGroupInformation.doAs(java.security.PrivilegedExceptionAction) org.apache.hadoop.mapred.Child.main(java.lang.String[])		orçoo
org.apache.hadoop.mapred.Child.main(java.lang.String[])		orc oi o c
		0 0 10 210
	100.00% 2545	.270sec.
<total></total>		
org.apache.hadoop.mapreduce.Mapper.run(org.apache.hadoop.mapreduce.Mapper\$Context) Inclusive: 91.73% 2334.763 sec. Exclusive: 0.10% 2.662 sec.		

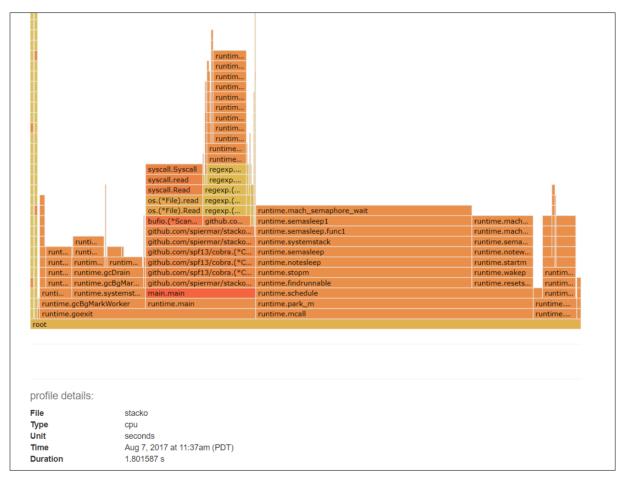
Source: https://www.oracle.com/technetwork/server-storage/solarisstudio/documentation/ o11-151-perf-analyzer-brief-1405338.pdf

Windows: PerfView (2017)



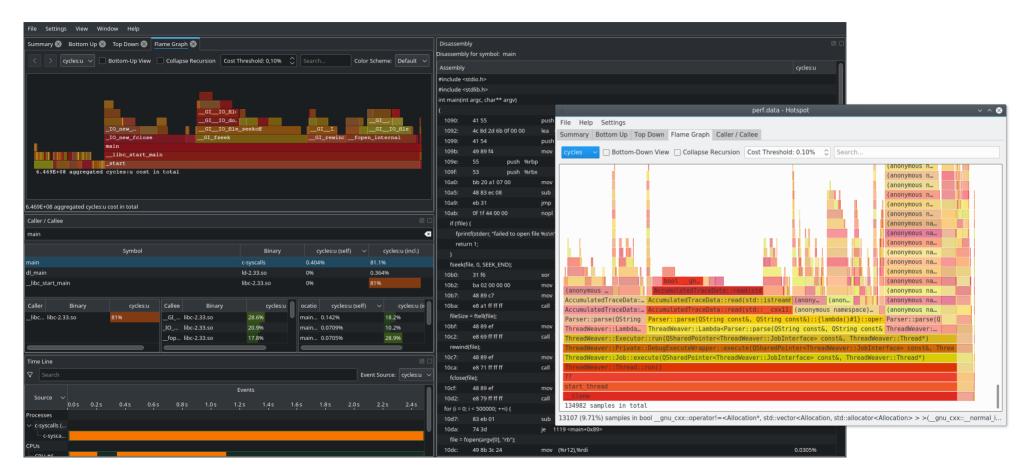
Source: https://github.com/Microsoft/perfview/pull/440 (Adam Sitnik)

Google: pprof (2017)



Source: https://github.com/google/pprof/pull/188 (Martin Spier)

Linux: hotspot (2017)



Source: https://github.com/KDAB/hotspot (Milian Wolff)

Eclipse Foundation: TraceCompass (2018)

Project Explorer 🛛 📃 🗖	🗄 Resources 🗔 Statistics 🗐		5:00:45.908	15:00:45.909	15:00:4	5 010	15:00:45.91	1	15:00:45.912	15:00:45.913	15:00:45.914	3 🖡 🗞 🏷 🔹 15:00:45.915	15:00:45.	
	Name		3.00.43.908	13.00.43.905	15.00.4	5.910	15.00.45.91		13.00.43.912	15.00.45.915	13.00.43.914	13.00.43.913	13.00.43.	
Tracing	Stack info not available (Is-traces,	/kerne												
Experiments [3]	v ls-traces/ust/uid/0/64-bit													
B+django [3]	▼ @ ls-26064													
ls-traces [2]	= main								main					
🎭 ls-traces/kernel							print_dir							
🛼 ls-traces/ust/uid/0/64-bit					<u> </u>		sort_files			print_current	_files			
v 🔤 Views							mpsort mpsort with t							
E Active Thread					mosort	with tmo	mpsor c_with_t	nsort into	tmp					
Context switch					mpsort_with m	psort_into	mpsort_wil	hmpsor	rt_with					
► ★ Counters					mps mps m	ipsmps	mpsmps	mp	mps					
CPU usage														
☆ Debug Info														
► Event Matching Latency														
Main Futex Contention Analysis														
Input/Output						0.00		- 66						
▲ IRQ Analysis						0.00		0.0	00					
Kernel memory usage					0.0.00	11 11 11		ii ii	í lí					
▲ Linux Kernel														
▶	🔲 Function Duration Statistic	cs 🕅					~ - 8	👌 Flam	ne Graph 🕱					$\downarrow^{a}_{\mathbf{Z}}\downarrow^{i}_{\mathbf{S}} ~ \bigtriangledown ~ \boxdot$
▶ 🔄 OS Execution Graph	Level	Minimum	Maximum A	Average Standa	rd Deviation	Count	Total			0.000 s	0.002 s	0.00	4 s	
▶	▼ ls-traces/ust/uid/0/64-bit			incruge sconor		count		▼ ls-260	64	1				
System Call Latency	▼ Total	161 ns	51.750 ms	4.799 µs	265.470 µs 1	111541	525 274 mc	0		_		main		
	md5 process block	249 ns	1.442 µs	268 ns	65 ns	5851	1.569 ms	1-		-		print_dir		
ontrol 🛛 🗖 🗖	iust	165 ns	924 ns	178 ns	44 ns	1	114.039 µs	2 —		filgobble_f	leprint_current	sort_fi	es	
	extract_dirs_from_files	3.619 µs	1	3.619 µs	44115	1	3.619 µs	3		xstr	print_file	mpso	t htmo	
M N % @ ►• 00 X ⊠	close stdout	71.929 µs		1.929 µs			71.929 μs	4			qu	mpsorc_with tmo	mosort into tmo	
e filter text 🛛 🕙	-						1.462 µs	6		_		mpsor mpsor	mpsort_with_tm	8
Local	heap_alloc	1.462 µs		1.462 µs 3.190 µs	_			7		_		mps)	mps mpso	
Provider	stat	3.190 µs					3.190 µs	8					mp	
Rernel	get_nonce	18.200 µs		B.200 µs	-	1	18.200 µs	9						
Sessions	quote_name	853 ns		1.064 µs	1.849 µs	379	403.253 µs	10 - 11 -						
363310113	version_etc_ar	1.970 µs		4.599 µs	-	2	9.199 µs	12 -						
	hard_locale	198 ns	302 ns	250 ns		2	500 ns	13 -		_			0.01	
	gobble_file	1.889 µs		2.155 µs	906 ns	380	818.734 µs	14 -				- 0.00 0.00		
	xrealloc	1.308 µs		7.199 µs		2	14.398 µs	15 -						
	default key compare	572 ns	572 ns	572 ns		11	572 ns							
	Els-traces 🛛 Els-traces	s/ust/uid/0/64	-bit											
	Trace	Timestamp		Channel	CPU Ever	nt type			Contents					
	🛷 <srch></srch>	<srch></srch>		<srch></srch>	<srch> <src< td=""><td>ch></td><td></td><td></td><td><srch></srch></td><td></td><td></td><td></td><td></td><td></td></src<></srch>	ch>			<srch></srch>					
	ls-traces/kernel	2018-01-29 1	5:00:45.917 659	9 277 kernel 2	2 rcu	utilizati	on		s=End context swi	itch, context, pe	rf cou migratio	ns=33		
	ls-traces/kernel	1	5:00:45.917 659			ed_stat_			comm=sort, tid=2	-				
	ls-traces/kernel		5:00:45.917 660		-	ed switc						prev_state=0, next_	comm=sort_nex	t tid=26070.nex
	ls-traces/kernel		5:00:45.917 661			call_exit			ret=326, buf=1407					
	and the cost in the cost of th								fd=3, offset=5567					
	ls-traces/kernel	2018-01-20.1												
	ls-traces/kernel ls-traces/kernel	2018-01-29 1	5:00:45.917 662 5:00:45.917 662			call_entr call_exit			ret=556795, conte					

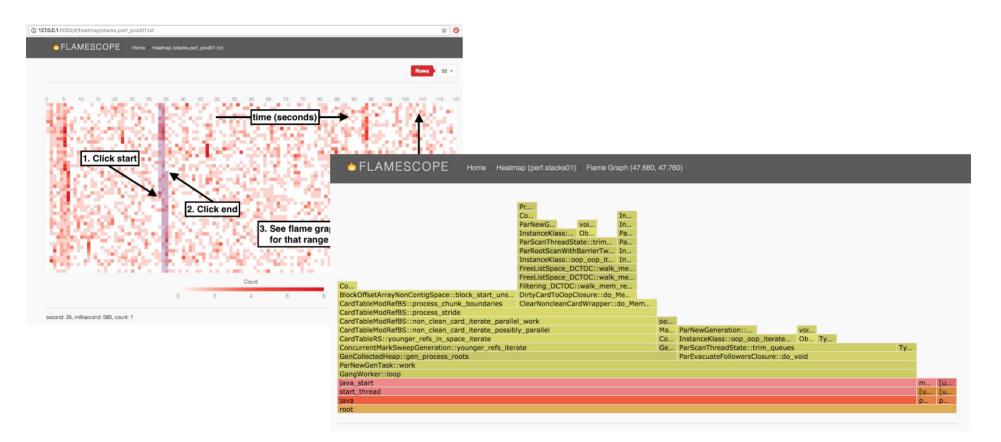
Source: https://www.eclipse.org/tracecompass/index.html

Java: Java Mission Control (2018)

6ª 🖓	Method Profiling 0		Execute	ExecuteThread.run()		ExecuteThrea				
Automated Analysis Results			Execute	ExecuteThread.execute(Runnable)		ExecuteThrea				
Java Application	<no selection=""> Select: <n< td=""><td></td><td></td><td></td><td></td><td></td></n<></no>									
Threads			SelfTuni	SelfTuningWorkManagerImpl\$WorkA	Provide Thermody and D	SelfTuningWo				
Wemory			ImageS	ImageSourceWork.run()	ExecuteThread.run()	ImageSource				
Cive Objects	Top Package		Instrum	InstrumentationImageSource.create	ExecuteThread.execute(Runna					
Dock Instances	weblogic.diagnostics.image.descriptor		BasicDe	BasicDescriptorManager.writeDescri	SelfTuningWorkManagerImpl\$	BasicDescript				
局 File I/O	weblogic.descriptor.internal		BasicDe	BasicDescriptorManager.writeDescri	ImageSourceWork.run()	BasicDescript				
Socket I/O	java.io		Marshall	MarshallerImpl.marshal(OutputStrea	InstrumentationImageSource.	MarshallerIm				
Co Method Profiling	java.util java.lang	Execute	Marshall	MarshallerImpl.marshal(XMLStream	Execute BasicDescriptorManager.writel	D MarshallerIm				
Exceptions	weblogic.xml.babel.baseparser	Execute	Marshall	MarshallerImpl.marshalBindingType(Execute BasicDescriptorManager.writel	D MarshallerIm				
Thread Dumps	sun.nio.cs	SelfTuni	PushMar	PushMarshalResult.marshalTopType(SelfTuni MarshallerImpl.marshal(Outpu	PushMarshalR				
JVM Internals	weblogic.xml.stax	ImageS	PushMar	PushMarshalResult.writeContents(Ru	ImageS MarshallerImpl.marshal(XMLS					
Garbage Collections	java.util.zip	Instrum	ByName	ByNameRuntimeBindingType.accept	Instrum MarshallerImpl.marshalBindin	and the second se				
GC Configuration	com.bea.staxb.runtime.internal	BasicDe	PushMar		BasicDe PushMarshalResult.marshalTo					
▶ Q. Compilations	weblogic.diagnostics.instrumentation.engine.bas	State of the second sec	Construction of the second second	PushMarshalResult.visit(ByNameRun		Contract Contract Contract Contract Contract				
G Class Loading	sun.misc	BasicDe	PushMar	PushMarshalResult.visitProp(Object,	BasicDe PushMarshalResult.writeConte					
Se VM Operations	java.util.regex	Marshall	PushMar	PushMarshalResult.marshalType(Obj	Marshall ByNameRuntimeBindingType.a					
TLAB Allocations	com.bea.objectweb.asm	Marshall	PushMarshalR	PushMarshalResult.writeContents(Runtim	Marshall PushMarshalResult.visit(ByNa.	Contrast Spaces Personal States of States and States				
G Environment	weblogic.diagnostics.instrumentation java.nio	Marshall	ByNameRunti	ByNameRuntimeBindingType.accept(Runti	Marshall PushMarshalResult.visitProp(C	ByNameRunti				
Processes	java.nio java.beans	PushMar	PushMarshalR	PushMarshalResult.visit(ByNameRuntimeB	PushMar PushMarshalResult.marshalTy	p PushMarshalR				
Environment Variables	weblogic.xml.babel.scanner	PushMarshalR	PushMarshalR	PushMarshalResult.visitProp(Object, Runti	PushMarshalResult.writeContents(Runtim	PushMarshalR				
System Properties		ByNameRunti	PushMarshalR	PushMarshalResult.marshalType(Object, R	ByNameRuntimeBindingType.accept(Runt	PushMarshalR				
Recording	Top Class	PushMarshalR	PushMarshalR	PushMarshalResult.writeContents(Runtim	PushMarshalResult.visit(ByNameRuntime					
Event Browser	weblogic.diagnostics.image.descriptor.Instrur	PushMarshalR	ByNameRunti	ByNameRuntimeBindingType.accept(Runti	PushMarshalResult.visitProp(Object, Runt	CONTRACTOR OF CONTRACTOR OF CONTRACTOR				
G1 Heap Layout		PushMarshalR	PushMarshalR	PushMarshalResult.visit(ByNameRuntimeB	PushMarshalResult.marshalType(Object, I					
Java FX		PushMarshalR	PushMarshalR.		PushMarshalResult.writeContents(Runtim					
19-Metadata				PushMarshalResult.visitProp(Object, Runti						
LG- Motoroto		ByNameRunti	PushMarshalR	PushMarshalResult.marshalType(Object, R	ByNameRuntimeBindingType.accept(Runt	Contraction of the local division of the loc				
		PushMarshalR	PushMarshalR	PushMarshalResult.writeStartElement(QN	PushMarshalResult.visit(ByNameRuntime					
		PushMarshalR	PushMarshalR	PushMarshalResult.writeStartElement(QN	PushMarshalResult.visitProp(Object, Runt					
		PushMarshalR	PrettyXMLStr	PrettyXMLStreamWriter.writeStartElement	PushMarshalResult.marshalType(Object, I	R PrettyXMLStr				
		PushMarshalR	XMLWriterBas	PrettyXMLStreamWriter.preStartElement()	PushMarshalResult.writeEndElement()	XMLWriterBas				
		PrettyXMLStr	XMLWriterBas	PrettyXMLStreamWriter.indent()	PrettyXMLStreamWriter.writeEndElement) XMLWriterBas				
		XMLWriterBas	XMLWriterBas	XMLWriterBase.writeCharacters(String)	XMLWriterBase.writeEndElement()	XMLWriterBas				
		XMLWriterBas	XMLWriterBas.	XMI WriterPass writeCharadersInternal/Ctring br	MLWriterBase.writeName(String, String,	String)				
		XMLWriterBase.wr		name: XMLWriterBase.writeCharacters(String), val						
		Writer.write(String	and the second se							
Properties 🖾 💩 Results		BufferedWriter.wri								
E 🕆 🗆 🕈 🔻	Click me to reset!									
operty Value						10.01.10				
	Stack Trace					K 3 K				
	Stack Trace			Count						
	1 void weblogic.diagnostics.image.descriptor.Instrumen	tationImageSourceBean	Impl.setInstrumentation	Events(InstrumentationEv 790						
	1 void weblogic.diagnostics.image.descriptor.Instrumen	tationImageSourceBean	Impl.addInstrumentation	Event(InstrumentationEv 790						
	1 InstrumentationEventBean weblogic.diagnostics.image.	descriptor.Instrumentatio	nimageSourceBeanimpl.o	reateInstrumentationEvent() 790						
	1 void weblogic.diagnostics.instrumentation.Instrument	ationImageSource.writei	RecentEvents(Instrumen	tationImageSourceBean) 790						
	t void weblogic.diagnostics.instrumentation.Instrument									
	t void weblogic.diagnostics.image.imageSourceWork.ru			790						
	1 void weblogic.work.SelfTuningWorkManagerImplSWor			790						
	1 void weblogic.work.ExecuteThread.execute(Runnable)			790						
	f void weblogic.work.ExecuteThread.execute(Rumable)			790						

Source: https://github.com/thegreystone/jmc-flame-view (Marcus Hirt)

Netflix: FlameScope (2018)



Source: https://netflixtechblog.com/netflix-flamescope-a57ca19d47bb (Brendan Gregg, Martin Spier)

Netflix: FlameCommander (2019)

	FLAMECOMN CPU Profile	MANDER Horne	Tools *			(+1258)	Q New CPU F	Help • Profile		
	All Profiles					O III 0	🕹 i 🗖	ompare		
	Date -	Instance / Container C	Account ¢	Reg	aion \$	ASG € ✓ FLAMECOMMANDER Home Tools •	Statu		Aller (a) Help +	
	04/19/2020→ 05/19/2020	×	×		Search Search X	Loon young apply interview and prevention control control control control control pre- Loon y network your matrix / large calls and when you be apply called a Loon you prevention / large called a set of the set of the set of the set of the Loon you prevention / large called a large called a set of the Loon you prevention / large called a large called a set of the Loon you prevention / large called a large called a set of the Loon you prevention / large called a large called a set of Loon you prevention / large called a large called a set of Loon you prevention / large called a large called a large called a set of Loon you prevention / large called a large called a large called a large called a Loon you prevention in the called a large called a large called a Loon you prevention (large called a large called a large called a Loon you prevention (large called a large called a large called a Loon you prevention (large called a large called a large called a Loon you prevention (large called a large called a large called a Loon you prevention (large called a large called a Loon you prevention (large called a large called a Loon you prevention (large called a large called a Loon you prevention (large called a large			Lcom/googi Lcom/googi Lcom/googi Lcom/googi Lcom/googi	pannybetrintern kr/governatorn je/injeetrintern je/injeetrintern je/injeetrintern je/injeetrintern je/injeetrintern
	May 19, 2020 11:36 AM	L0ffcef2f4e34a604d		eu	Settings Flavor	Loonly appearing inject internally injection many, inclusion and inclusions and i	Loom Loom Loom Loom Loom		Lcontry (goog) Lcontry (goog) Lcontry (goog) Lcontry (goog) Lcontry (goog)	ple/inject/intern ple/inject/intern tt/GeneratedMe ple/inject/intern ple/inject/intern
CPU Profile > i-	May 19, 2020 11:33 AM	i-06817e7970cf26f03		au	Standard Inverted Merge Balance Profiles Demangle	Lcom / google / inject/internal/ProvisionListenerRackCallbackSP Lcom / google / inject/internal/ProvisionListenerRackCallbackSP Lcom / netFix/governator/event/ApplicationLiventModuleSAppliC Lcom / netFix/governator/LiveryCalListenerModuleSLiVeryCalList. Lcom / netFix/governator/LiveryCalListenerModuleSLiVeryCalList. Lcom / netFix/governator/LiveryCalListenerModuleSLiVeryCalList.	Lcom Lcom Lcom Lcom		Lcorr(netTi) Lcorr(googi Lcorr(netTio Lcorr(netTio Lcorr(netTio Lcorr(netTio	ple/inject/intern ix/governator/ ik/governator/ ik/governator/ ple/inject/intern ix/governator/ ple/inject/intern
Settings	GooAPTRE147	pid #13547 (java)		us	C Certangle	Lcom/netflix/governator/LifecycleModule\$LifecycleProvisionLis. Lcom/google/inject/internal/ProvisionListenerStackCaliback\$Pr Lcom/google/inject/internal/ProviderMethod\$Factory::get Lcom/google/inject/internal/ProviderToInternalFactory.cgater. Lcom/google/inject/internal/ProviderToInternalFactory.cgater.	Lcom Lcom Lcom/	i.	Lcom/googl Lcom/googl Lcom/googl	jle/inject/intern jle/inject/intern jle/inject/intern jk/nfcontext/im
Break By PID Bows 10 Enhanced	•	10 10 20 10 20 10 10 10 10 10 10 10 10 10 10 10 10 10	SKC.		How to interpret differential flame graphs • Otherential flame graph is identical to the time flame graph (has the same shape and sample ocurd). • It uses counts to highlight the differences between time and	Long/papping/apping/inget/interact/inget/inget/apping/appi	Ljava/ Ljava/ Ljava/ gava/ gava/ gava/ toom/ gava/ toom/ gava/ tja tjava/ tja tja tjava/ tjava/ tja tjava/ tjava/ tjava/ tja tjava/	Loo Loo Loo Loom/ Loom/ Loom/	Loom/special Loom/special Loom/special Loom/setTin Loom/setTin Loom/setTin Loom/setTin Loom/setTin Loom/setTin Loom/setTin Loom/setTin	je/inject/intern je/inject/intern je/inject/intern ik/gs2/group/generat ix/gs2/group/generat ix/gs2/group/generat ix/seededbinding/Se ix/seededbindings/Se ix/seededbindings/Se ix/seercalls\$Unan ix/grpc/interceptor/ie
		Let a	1975		It frames mean less samples in the first lane graph; If the first lane graph lane of difference between first lane graph lane of the first lane graph lane of the first lane graph lane of the first lane lane graph. Unto in first lane lane lane lane lane lane lane lane	Loss Hearthurg part Hearthurg Hospital, Context J. Copyright Carlos Sale with Hearthurg Lob March 1999 (2014). The second	r\$11100048/IClose Italianer-most48/IClose (65.007%, 1 \$1\$21100488/IClose nHal/Close nHal/Close Hal/Close Hal/Close	Unx/Int_	Loom/net/file Loo/proc/file Loom/net/file Loom/net/file Loo/proc/for Loo/proc/for Loo/proc/for Loo/proc/for Loo/proc/for Loo/proc/for Loo/proc/for	ix/grpc/interceptor/lo prwardingServerCall. ix/grpc/interceptor/n ix/concurrency/limits ontexts\$Contextualiz prwardingServerCall. prwardingServerCall. ternal/ServerCall. ternal/ServerTall. ternal/ServerTall.

Source: https://www.youtube.com/watch?v=L58GrWcrD00 (Martin Spier, Jason Koch, Susie Xia, Brendan Gregg) 24

AMD: uProf (2019)



Source: https://developer.amd.com/amd-uprof/?sf215410082=1

Java: YourKit (2019)

CP	U × I Threads × Memory ×	🙆 Ga	arbage Objects × 🛛 🗲 Excepti	tions * 🖄 Performance Charts * 👁 Events * 👁 Inspections * 🗎 Summary *
	CPU profiling Tracing		XMLDocum SchemaPar SchemaPar SchemaDO	XMLDocum SchemaPar: SchemaPar:
\succeq	CPU usage telemetry		XSDHandle \Leftrightarrow XSDH: \Leftrightarrow \Leftrightarrow XSDHandler.parseSchem XS	XSDHandle + + + + + + + + + + + + + + + + + + +
٦*	Call tree – All threads merged		XMLSchemaLoader.load: XSCc XMLSchemaLoader.loadGramm	XMLSchemaLoader.loa
۲ ۲	Call tree – By thread		XMLSchemaLoader.loadGramm	XMLSchemaLoader.loa
1	Flame graph		SchemaFactory.newSchema	SchemaFactory.newSch 🕈 SchemaFactory.ne 🕈 SchemaFactory.n 🕈 SchemaFactory
Ŵ	Hot spots			LocalXsdResolver.resolv LocalXsdResolver.resolv LocalXsdResolver.resolv LocalXsdResolver.buildb LocalXsdResolver.buildb LocalXsdResolver.buildb
≣	Method list		MappingXsdSupport. <init> MappingXsdSupport.<clinit></clinit></init>	
	Java EE		JpaOrmXmlEventReader.mapNat JpaOrmXmlEventReader.mapNat	
-	Database		JpaOrmXmlEventReader.mapNar JpaOrmXmlEventReader.wrap	amespaces STA
\bigcirc	JSPs and servlets		JpaOrmXmlEventReader.wrap JpaOrmXmlEventReader.peek	org.hibernate.boot.jaxb.internal.stax.JpaOrmXmlEventReader.wrap(StartElement)
Ω	JNDI		MappingBinder.toDom4jDocur	Time (ms): 539 6 % Avg, Time (ms): 539
			mappingomacriaoonia	Count: 1 Double click to zoom
			InputStreamXmlSource.doBind	

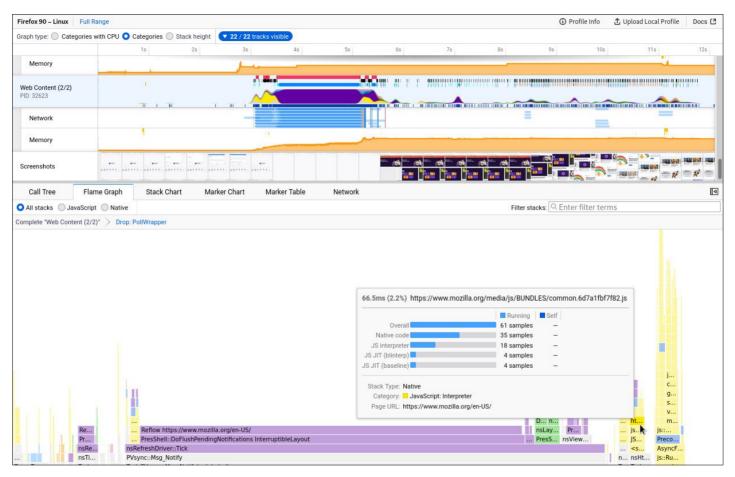
Source: https://www.yourkit.com/docs/java/help/cpu_flame_graph.jsp

Java: IntelliJ IDEA (2019)

Prof	iler: 📄 [Async Profiler]	Memory 54646 Scheduler $ imes$ \blacksquare Java Flight Recorder 54666	δ Scheduler × <mark>-</mark> τ≣1 φ —							
Ľ	Sampling Call Tree Ever	S								
	Flame Graph Call Tree	Methods List								
	All threads merged	java.lang.invoke.DirectMethodHandle.make i.l.i.DirectMethodHandle.make								
	JFR request timer id=6 main id=15	j.l.i.DirectMe java.lang.invoke.LambdaForm.createlo j.l.i.DirectMe 1 samples, 100.00% of parent, 33.33% of								
		i:1 java.lang.invoke.LambdaForm. <clinit></clinit>								
		java.lang.invoke.LambdaForm.createldentityForms java.lang.invoke.BoundMethodHandle. <clinit></clinit>	.Scheduler.main java.lang.invoke.MethodHandl							
		Q j.l.i.BoundMethodHandle\$SpeciesData. <clinit> j.l.i.BoundMethodHandle\$SpeciesData.initForBootstrap</clinit>	java.lang.invoke.MethodHandl java.lang.invoke.CallSite. <clinit< td=""></clinit<>							
		j.l.i.BoundMethodHandle\$Factory.makeCtors	java.lang.invoke.MethodHandl							
		j.l.i.BoundMethodHandle\$Factory.makeCbmhCtor java.lang.invoke.MethodHandles\$Lookup.findStatic	j.l.i.MethodHandles\$Lookup.g							
▶_4	► 4: Run \approx 6: TODO C Profiler Terminal Services									

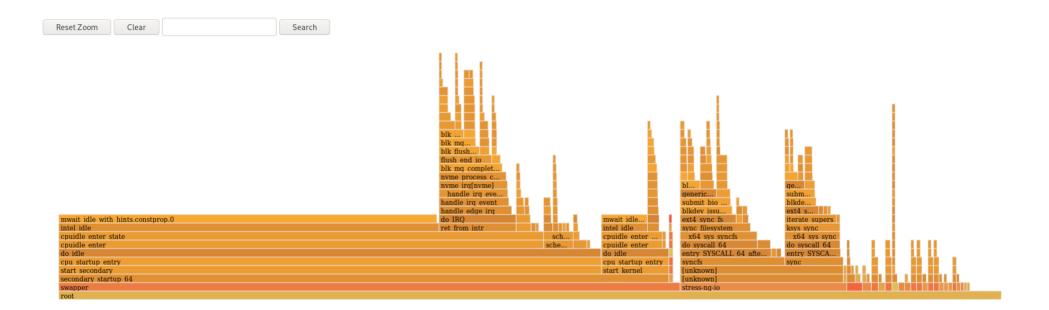
Source: https://blog.jetbrains.com/idea/2019/06/intellij-idea-2019-2-eap-4-profiling-tools-structural-search-preview-and-more/

Firefox: Profiler (2019)



Source: https://profiler.firefox.com

Linux: perf script flamegraph (2020)



Source: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/ monitoring_and_managing_system_status_and_performance/getting-started-withflamegraphs_monitoring-and-managing-system-status-and-performance (Andreas Gerstmayr)

MathWorks: MATLAB Profiler (2020)

📣 Profiler				-	
PROFILER					
Print Profile Province Forward File NAVIGATE SEARCH	Highlight ([preypeaks,predatorpeaks]	= solvelotka(0,15,[2 PROFILE	20;20]) • p	Run and Time
Profile Summary (Total Time: 0.052	s)				
✓ Flame Graph					
	_		isLocalE	xtrema xtrema>isLoca isLo	
odeget funfun\private\odearguments ode23 solvelotka Profile Summary Generated 12-Nov-2019 11:22:04 using performanc	e time.		isLocalE isLocalE islocalm	xtrema>isLoca isLo xtrema	
funfun\private\odearguments ode23 solvelotka Profile Summary	e time. Calls	Total Time (s) ↓	isLocalE isLocalE islocalm	xtrema>isLoca isLo xtrema ax	
funfun\private\odearguments ode23 solvelotka Profile Summary Generated 12-Nov-2019 11:22:04 using performanc		Total Time (s) ↓ 0.052	isLocalE3 isLocalE3 islocalm solvelot	xtrema>isLoca isLo xtrema ax ka>calculatepeaks Total Time Plot	
funfun\private\odearguments ode23 solvelotka Profile Summary Generated 12-Nov-2019 11:22:04 using performanc Function Name	Calls		isLocalE3 isLocalE3 islocalm solvelotA	xtrema>isLoca isLo xtrema ax ka>calculatepeaks Total Time Plot	
funfun\private\odearguments ode23 solvelotka Profile Summary Generated 12-Nov-2019 11:22:04 using performance Function Name solvelotka	Calls	0.052	isLocalE3 isLocalE3 islocalm solvelot	xtrema>isLoca isLo xtrema ax ka>calculatepeaks Total Time Plot	

Source: https://www.mathworks.com/help/matlab/matlab_prog/profiling-for-improving-performance.html

AWS: CodeGuru (2020)

ImageProcessingWebApp-Ar	omaly Info	3 Recommendations Actions
Overview v Latency v	Q Search for frame	2020-06-26 @ 21:55 - 22:45 BST last 1
Legend BLOCKED NATIVE RUNNABLE TIMED_WAITING WAITING	LCMSTransf Deflater. LCMST LCMSTra Deflater. Deflater. LCMSTransf IDATOutput LCMST LCMSTra IDAT NATIVE: a year ColorComve IDATOutputS Color ColorCo IDAT NUNNABLE: 2 months JPEGImageR PNGmageWrit JPEGI JPEGImag PNG 10.22% of total time JPEGImageRe PNGmageWrit JPEGImag PNG Estimated active CPU of JPEGImageRe kmageWriter. JPEGImag ImageIO. ImageIO. ImageIO.rea ImageIO.doWr ImageIO. ImageIO. ImageIO.	
	ImageProcessor\$BrightenIm ImageProcessor\$Darkenim ImageProcessor\$Darkenim ImageProcessor	
	ImageProcessor\$Darkenim ImageProcessor\$Darkenim ImageProcessor ImageProcessorsDarkenim	SGr ImageProcessor\$Up AmazonSQSClien TimeUnitisleep Main.createMessageToImageQueue
	ImageProcessorSLambda.run	MainSLambda.run
Unsafe.park	ExecutorsSRunnableAdapter.call	
LockSupport parkNanos	FutureTaskirun	
AbstractOueuedSynchronizer\$Condit	ThreadPoolExecutor.runWorker	

Source: https://aws.amazon.com/codeguru/features/

Google: Cloud Profiler (2020)

1 Focus : compress/flate.(*huffmanBitWriter).write Add profile data filter... × 0 Metric : CPU time 🔞 + Ŧ (*Writer).Flush (*Writer).Flush (*compressor).syncFlush (*compressor).deflate (*compressor).writeBlock (*huffmanBitWriter).writeBlock (*huffmanBitWriter).writeBytes (*huffmanBitWriter).write: 88.16 ms (2.80%), averaged over 250 profiles (*Buffer).Write (*Buffer).grow makeSlice nemclrNoHeapPointers

Source: https://cloud.google.com/profiler/docs/focusing-profiles

Intel: vTune (2021)

User	Syste	m Synchronizat	📃 Overhead	Other		🛃 57 K K 🕅	natvec Q
	miniF mini		niFE::CSRMatrix <do< td=""><td>uble, int, int>, miniFE::Vector<double, int="" int,=""></double,></td><td></td><td></td><td></td></do<>	uble, int, int>, miniFE::Vector <double, int="" int,=""></double,>			
	[OpenMP [Op	[OpenMP dispatcher]			mi		
	kmp_fork	kmp_fork_call			[0		
[Op	[OpenMP f [Op	[OpenMP fork]				k	
mini	miniFE::da mini	miniFE::matvec_std <mi< td=""><td>niFE::CSRMatrix<do< td=""><td>uble, int, int>, miniFE::Vector<double, int="" int,=""></double,></td><td></td><td></td><td></td></do<></td></mi<>	niFE::CSRMatrix <do< td=""><td>uble, int, int>, miniFE::Vector<double, int="" int,=""></double,></td><td></td><td></td><td></td></do<>	uble, int, int>, miniFE::Vector <double, int="" int,=""></double,>			
mini	miniFE::cg_solve <min< td=""><td>iFE::CSRMatrix<double, int,="" int<="" td=""><td>, miniFE::Vector<do< p=""></do<></td><td>ouble, int, int>, miniFE::matvec_std<minife::c< td=""><td>SRMatri mi</td><td>nkmp_fork_barrier</td><td></td></minife::c<></td></double,></td></min<>	iFE::CSRMatrix <double, int,="" int<="" td=""><td>, miniFE::Vector<do< p=""></do<></td><td>ouble, int, int>, miniFE::matvec_std<minife::c< td=""><td>SRMatri mi</td><td>nkmp_fork_barrier</td><td></td></minife::c<></td></double,>	, miniFE::Vector <do< p=""></do<>	ouble, int, int>, miniFE::matvec_std <minife::c< td=""><td>SRMatri mi</td><td>nkmp_fork_barrier</td><td></td></minife::c<>	SRMatri mi	nkmp_fork_barrier	
miniFE::	driver <double, int="" int,=""></double,>	•				kmp_launch_thread	
main							_src_z_Linux_util_cpp_cfe967e4
libc_st	tart_main					start_thread	
_start						clone	
Tel							
Matched:	: 1062.158s of 1703	3 590s (62 3%)					
imatcheu.	. 1002.1005.01 1700	0.0003 (02.078)					

Source: https://www.intel.com/content/www/us/en/develop/documentation/vtune-help/top/reference/ user-interface-reference/window-flame-graph.html

Splunk: AlwaysOn Profiling flame graph (2021)

≡ splunk> Observability											Q,	+ [
-1h * Not Specified * Service: birdlympics2 *												Clear
AlwaysOn Profiling												
Q. Search stack frames	0.0	Jsing 7	0.2k call stacks	out of 70.6k total. L	oad More			[Table	Both	Fla	ime Grap
Total												
Thread.run(java.base@11.0.13/Thread.java:829) TaskThread\$WrappingRunnable.run(TaskThread.java:61)				FastThreadLocalRun		ThreadPoolExec						Selec.
ThreadPoolExecutor\$Worker.run(jeva.base@11.0.13/ThreadPoolExecutor ThreadPoolExecutor.runWorker(java.base@11.0.13/ThreadPoolExecutor.			ThreadPoolEx		cutorMap\$2.run(Threa ThreadPoolExecutor.ru adEventExecutor\$4.ru ThreadPoolEx Threa		NIDER		Threa	Three.	Selec	
ThreadPoolExecutor getTask(jsva.base@11.0.13/ThreadPoolExecutor.jav				EpollEventLoop.run	FoollEvent	Synchronous	Sche	CONTRACTOR OF THE OWNER	EPoll	Sche		EPoll.
TaskQueue.take(TaskQueue.java:33)	Scheduled		NioEndpoint\$	EpollEventLoop.ep	dependence of the local division of the loca	Synchronous	Sche.	Serve		Sche		and the second
TaskQueue.take(TaskQueue.java:108)	Support and the owner of the owner own			Native.epol/Wait(N	I TANK INCOME IN CONTRACTOR	Synchronous	Abstr			Abstr.		
LinkedBlockingQueue.take(java.base@11.0.13/LinkedBlockingQueue.j	Abstr. A	bstr.	AbstractProc	Native epol/WaltOC	Native.epo	LockSupport	Lock			Lock		
AbstractQueuedSynchronizer\$ConditionObject.await(jova.base@11.0	Lock Le	ock	Http://Proces		Native.epo	Unsafe.park(ja	Unsaf			Unsaf		
LockSupport park(java.base@11.0.13/LockSupport java:194)	Unsaf U	nsaf	CoyoteAdapte									
Unsafe.park(java.base@11.0.13/Native Method)			StandardEngi									
			ErrorReportVa									
			StandardHost									
			Authenticator									
			StandardWra									
			ApplicationFilt									
			ApplicationFilt									
			OncePerRegu.									

Source: https://docs.splunk.com/Observability/apm/profiling/using-the-flamegraph.html

New Relic: flame graphs (2021)

New Relic ONE**				~ ×
Services	flamegraph-service (staging)			
flamegraph-s		41		D Notes for all
	8f974d6c22ff/172.17.0.20 / from Nov 2, 02:36 pm to Nov 2, 02	41 pm 🗸		No logs found
Summary				
		AVERAGE CPU AVERAGE HEAP M 11.2% 247.03 MB	IEMORY TOTAL GC PAUSE TIME HEAP USED 0.5 Sec 245.46 MB	HEAP SIZE HEAP COMMITTED 2,048 MB 1,409 MB
		11.2% 247.03 WB	0.5 Sec 245.40 IVIB	2,048 1016 1,409 1016
MONITOR				
Distributed tracing	java.lang.Thread.run()V:834 (count: 1990)			
Distributed tracing	java.util.concurrent.ThreadP org.eclipse.jetty.util.thread.QueuedThreadPool\$Run		sun.rmi.transport.tcp.TCPTransport\$AcceptLoop.run()\	
Service map	java.util.concurrent.ThreadP org.eclipse.jetty.util.thread.QueuedThreadPool.runJ sun.rmi.transport.tcp.TCPTr org.eclipse.jetty.server.AbstractConnector\$Acceptor		sun.rmi.transport.tcp.TCPTransport\$AcceptLoop.execu sun.management.jmxremote java.net.ServerSocket	neAcceptLoop()V:394 (count: 746) t.accept()Ljava/net/Socket;:533 (count:
	java.security.AccessControll org.eclipse.jetty.server.AdstractConnector.Accept())V:			t.implAccept(Ljava/net/Socket;)V:565 (c
Dependencies	sun.rmi.transport.tcp.TCPTr sun.nio.ch.ServerSocketChannelImpl.accept()Ljava			SocketImpl.accept(Ljava/net/SocketImp
Transactions	sun.rmi.transport.tcp.TCPTr sun.nio.ch.ServerSocketChannelImpl.accept(Ljava/			npl.socketAccept(Ljava/net/SocketImpl;
	sun.rmi.transport.tcp.TCPTr sun.nio.ch.ServerSocketChannelImpl.accept0(Ljava	/io/FileDe org.eclipse.jetty.util.thread.st org.eclipse.jetty.util.thread.st	java.net.PlainSocketImpl.soc	
Databases	sun.rmi.transport.tcp.TCPTr	org.eclipse.jetty.io.Managed org.eclipse.jetty.io.Managed		
External services	java.io.FilterInputStream.rea	org.eclipse.jetty.io.Managed org.eclipse.jetty.io.Managed		
External services	java.io.BufferedInputStream	sun.nio.ch.SelectorImpl.sele sun.nio.ch.SelectorImpl.sele		
JVMs	java.io.BufferedInputStream java.net.SocketInputStream	sun.nio.ch.SelectorImpl.lock sun.nio.ch.SelectorImpl.lock sun.nio.ch.EPollSelectorImpl sun.nio.ch.EPollSelectorImpl		
	java.net.SocketinputStream	sun.nio.ch.EPoll.wait(IJII)1:-1 sun.nio.ch.EPoll.wait(IJII)1:-1		
Threads				
		Marking Children W		
EVENTS	User CPU Usage % Since Nov 2, 02:36 pm until Nov 2, 02:41 pm	Machine CPU Usage % Since Nov 2, 02:36 pm until Nov 2, 02:41 pm	GC Heap Sizes Since Nov 2, 02:36 pm until Nov 2, 02	:41 pm
Errors	0.06	16	2 k	
Violations	0.05		1.5 k	
violations	0.04		1.0 K	
Deployments	0.03	8	1 k	
	0.02	6		
Thread profiler		4	500	
	0	0		
REPORTS	02:36:00 PM 02:37:00 PM 02:38:00 PM 02:39:00 PM 02:40:00 PM 02:4	02:36:00 PM 02:37:00 PM 02:38:00 PM 02:39:00 PM 02:40:00 PI		:38:00 PM 02:39:00 PM 02:40:
			The same second of the second of the second se	Service in Service in Service
SLA	CPULoad.jvmSystem CPULoad.jvmUser	CPULoad.Total	Heap Used Heap Free	

Source: https://docs.newrelic.com/whats-new/2021/07/whats-new-july-8-realtime-profiling-java/

DataDog: profiling flame graph (2021)

•	product-re	rofiler ecommendatio Analysis (4)			보 Download Profile Data			
4	Analysi	is: The program	generated 26,255 exceptions per second during 4/29/2021, 7:	54:01 PM – 7:54:59 PM. This could impact performance.		View Thrown Exceptions Profile	View Full Analysis	
	ype CPU T	lime :	2m 34s 🔻 🍸 Filter flamegraph by method or package		Options	🗹 Method 👻		
n		Frame width represents the CPU time per method (Ctrl + Scroll to zoom)					1m 5s	
	PU Time: 2m	Time: 2m 34s over 58.37s, across all threads					ModelTraining.computeCoeffici 60s	
	м	Thread.run()	Thread.run()			ScopeInterceptor\$Delegatin 11.24s		
	arbage C	ThreadPoolExecutor\$Worker.run()				GenCollectForAllocation	5.365	
••		ThreadPoolExecutor.runWorker(ThreadPoolExecutor\$Worker)				GenCollectFull	5.22s	
0			re\$AsyncRun.run()	CompletableFuture\$AsyncSupply.run()				
(4)		GenerateRecomm	nendations.run() nGraph.populate()	ModelTraining.lambda\$computeCoefficients\$0(Tracer, int, List, int, int, CountDownLa ModelTraining.computeCoefficientsOnRange(List, int, int)	ModelTrain	HandshakeOneThread	825ms	
			iGraph.populate() iGraph.populate(int, ProductNode)	ModelTraining.computeCoefficientsOnRange(List, Int, Int) ModelTrainingsFeature)	BaseDecor	AbstractPipeline.wrapAnd		
цц.			Graph.populate(int, ProductNode)	modernaming.computecoentcientsoni eatare(modernamingsreatare)	ScopeInter	ThreadLocal\$ThreadLocal\$	Ma 587ms	
		RecommendationGraph.populate(int, ProductNode) RecommendationGraph.populate(int, ProductNode)			Scoperiterin	ThreadLocal\$ThreadLocal\$	Ma 585ms	
						ThreadLocal\$ThreadLocal\$	Ma 417ms	
						ExceptionHistogram.record	d(395ms	
ē					1	ConcurrentHashMap.com	ou 359ms	
\odot					Í	LongAdder.increment()	333ms	
V						ThreadDump	332ms	
٢		RecommendationGraph.populate(int, ProductNode)				ProductNode.removeChild	(P 303ms	
		Product.<	RecommendationGraph.populate(int, ProductNode)			RecommendationGraph.pd		
		Product.l	Product. <init>(int, String, boolean)</init>					
			Product.loadAssets(int)			Product. <init>(int, String, b</init>		
						Throwable.fillInStackTrace		
						StreamingSampler\$Counts	s.a 227ms	
0						SuperList.lambda\$remove	\$0 217ms	
						ThreadPoolExecutor.runW	or 216ms	
22						BufferedInputStream.read	() 214ms	
				1	0	ReferencePipeline\$2\$1.acc	e 204ms	
						AbstractDinalina wranCinki	Ci 172mc	

Source: https://www.datadoghq.com/knowledge-center/distributed-tracing/flame-graph/

Granulate: gprofiler (2022; now Intel)

java											
[/liso]	[/lib/x82.23.so]	[u]] java/lan;	g/Thread.run()V	/				start_thread		/8/
[/uso]	[] [/usrm.so]		io/n()\		we/util/consurrent/ThreadDeclEveruterfM/orkersun()V	org/apache/cas	sandra/concurrent/SEPWorker.r	un()V	/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so	[]	107
[/uso]	[] [/usrm.so]		io/nn(il/concurrent/SingleThreadEventExecutor\$2.run()V	org/aprun()V	org/apache/careTask.run()V	org/n()V		[]	10
	[] [/usrm.so]		io/nn(V Total time	3.23% Own time 0% Samples 15605 n()V	java/ubject	org/apache/careTask.run()V	javaJ)V		(10
[/uso]	[] [/usrm.so]		i	ioV	org/apache/spark/schedutem)Ljava/lang/Object	io/netun()V	java/util/conc/lang/Object	sun/J)V	/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so	(III	liii
[/o]	[] [/usso]		i	ioV	o org/apache/spark/sch)Ljava/lang/Object	io/netect)V	orV org/apachele.run()V	/usrso	/usr/lib/jym/java-8-openjdk-amd64/jre/lib/amd64/server/libjym.so		11P
[/]	[[/usso]		i	ioV	o org/apache/spark/schscheduler/MapStatus	io/netct)V	org/apachyThrow()V	it	/usr/lib/jvm/java-8-openjdk-amdjre/lib/amd64/server/libjvm.so		1V
[[[/uso]			i]	org/apator o org/apacheerator)V	io/netct)V	o org/aparator		/usr/lib/jvm/java/server/libjvm.so		1 P
[[/uo]			i]	org/apator o org/apacrator)V	org/apct)V	o org/aprator		/usr/lib/jvm/jver/libjvm.so		4P
	[/uo]			iV	org/apator j o scalaject	org/apst)V	oV org/aprator		/usr/libjvm.so		æ
	[/uo]			oV	org/apator j o co	org/aponse	oV org/aprator		/usrm.so		ΔĨ
	[/]	11		i]	org/apator j co	org/apsage	o org/or		/Juso		1Ĩ
				iV	org/aator jV co	org/apsage	oV orgor		7	111	/II
iii				iV	org/aator jV	org/apsage	orgor			i inv	/II
				i	org/ator j	org/asage	orgor			1 1111/	1
			1	iV	org/tor j	org/aRows	oV				/II
			- 1	iV	org/tor jV	org/ery				110	41
			- i	i]	org/tor jV	org/ery				i ini	1
			- i	iV	orgor j	org/ist				110	11
			- 1	iV	sca)V j	org/ist				117	
			i	0	sca)V jV	orgt					
			- i	0	scat	orgt				111	
			- ii		orgt	orgt					
			- ii		orgr	org)V					
			- i		orgr	or					
			- i		orgr	jaV					
			- ii		or]	jaV					
			- i i		or	i.V					
					or]	i V					
					orV						
					0						
					0						
					O						
11											
/a 🔴	Python 😑 PHP	• C++	left Kernel	🔵 Go 🕚 O	Other						

Source: https://docs.gprofiler.io/about-gprofiler/gprofiler-features/views/flame-graph

Microsoft Visual Studio: Flame Graph (2022)

CL	rrent View: Flame Grap	oh 🗸 🔍 Reset Zoom 🛣 Flip Flame Graph									
		DomainBoundILStubClass.IL_STUB_COMtoCLR(long)									
	BookService.WebApiApplication.Application_Start()										
	System.Web.Mvc.Are	eaRegistration.RegisterAllAreas(System.Web.Routing.RouteCollection, System.Web.Htt									
C	\Users\mihavs\source\	repos\BookService\BookService\Global.asax.cs:16									
	(0) (1) (1111 (1) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0	11 {									
		12 = public class WebApiApplication : System.Web.HttpApplication									
		13 {									
		14 🗄 protected void Application_Start()									
		15 {									
	588 (10.20%)	16 AreaRegistration.RegisterAllAreas();									
	116 (2.01%)	17 GlobalConfiguration.Configure(WebApiConfig.Register);									
	3 (0.05%)	<pre>18 FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);</pre>									
		19 RouteConfig.RegisterRoutes(RouteTable.Routes);									
	27 (0.47%)	20 BundleConfig.RegisterBundles(BundleTable.Bundles);									

Source: https://learn.microsoft.com/en-us/visualstudio/profiling/flame-graph

GrafanaLabs: Grafana flame graph (2022)



Source: https://grafana.com/docs/grafana/next/panels-visualizations/visualizations/flame-graph

Flame Graph Adoption

Implementations: >80

Related open source projects: >400

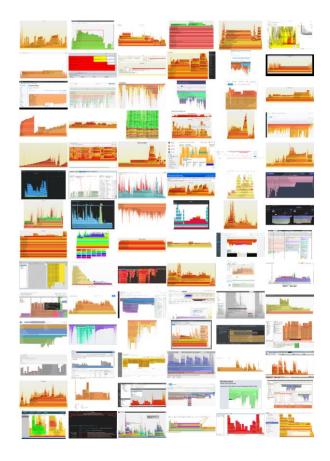
Commercial product adoptions: >30

New startups: 4 (so far)

Startup exits: 1 (so far)

Industry investment: >AUD\$1B

End users: ? (a lot)



2. CPU PROFILING

An Introduction to Flame Graphs

Stack Traces

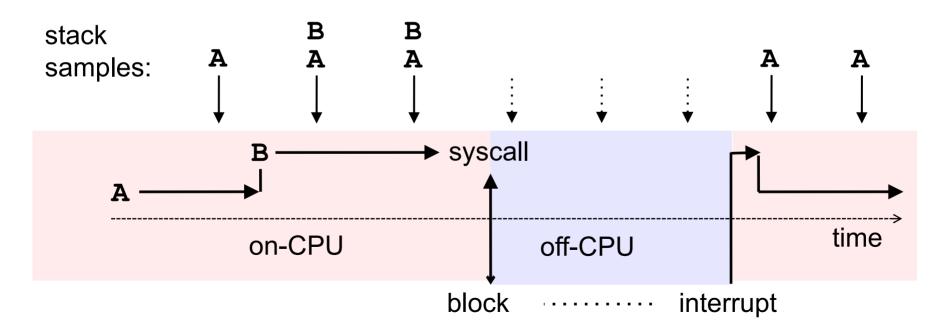
A code path snapshot. e.g., from jstack(1):

```
$ jstack 1819
[...]
"main" prio=10 tid=0x00007ff304009000
nid=0x7361 runnable [0x00007ff30d4f9000]
  java.lang.Thread.State: RUNNABLE
   at Func abc.func c(Func abc.java:6)
                                                  running
   at Func abc.func b(Func abc.java:16)
                                                  parent
                                                  g.parent
   at Func abc.func a(Func abc.java:23)
                                                  g.g.parent
   at Func abc.main(Func abc.java:27)
```

CPU Profiling

Record stacks at a timed interval

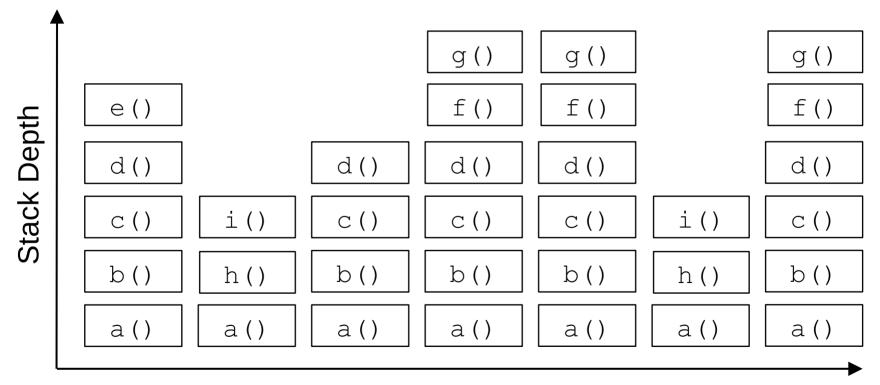
- Pros: Low (deterministic) overhead
- Cons: Coarse accuracy, but usually sufficient



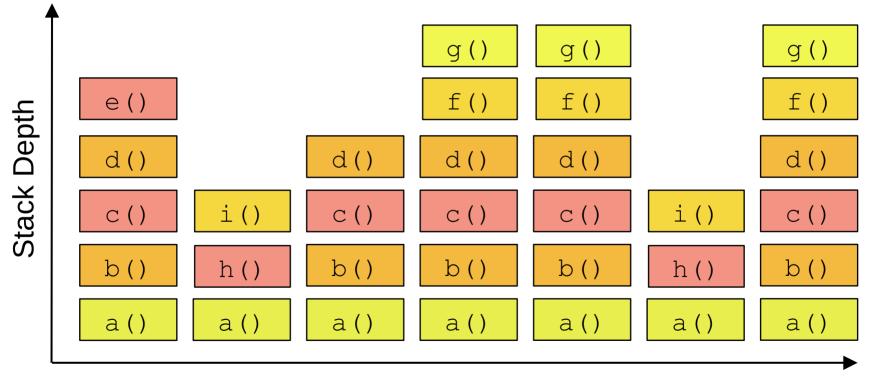
Stack Samples

	a()	a()	a()	a()	a()	a()	a()
S	b()	h()	b()	b()	b()	h()	b()
itack	с()	i()	C ()	C ()	С()	i()	С()
Stack Depth	d ()		d ()	d ()	d ()		d ()
oth	e()			f()	f()		f()
-				g()	g()		g()

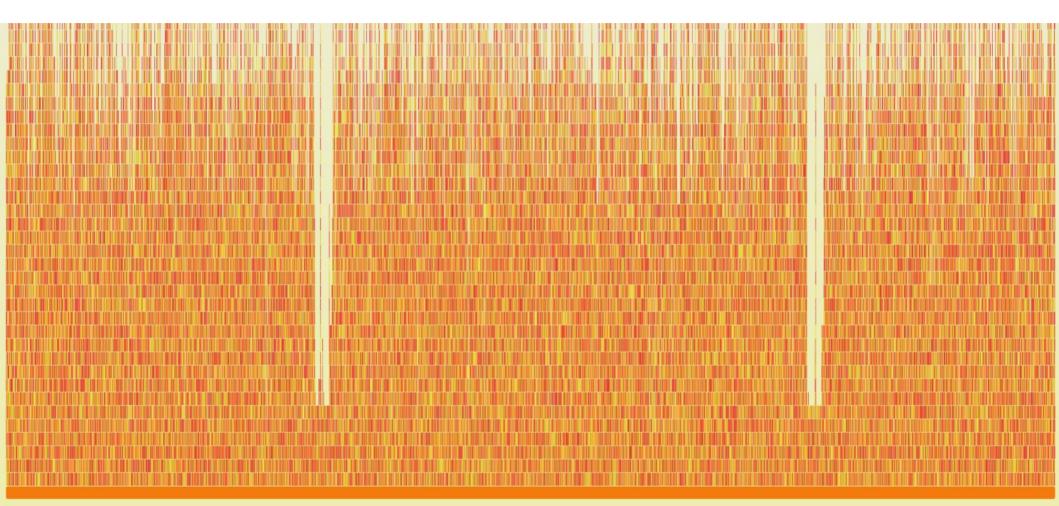
Stack Samples



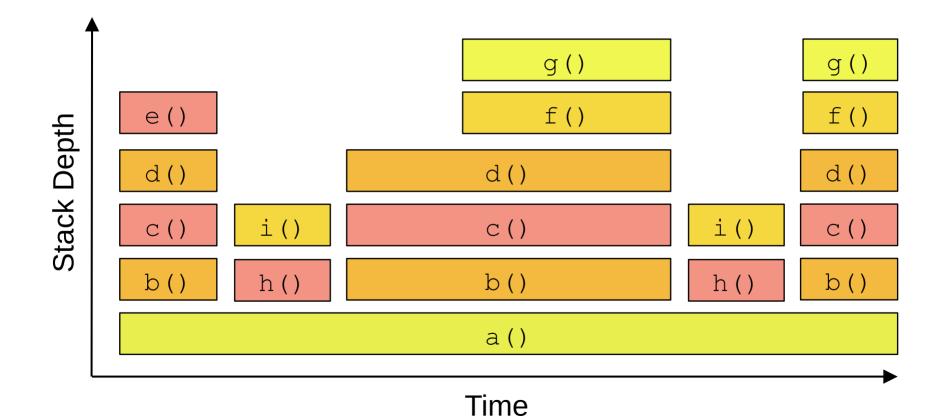
Stack Samples



Example Profile ("hair graph")



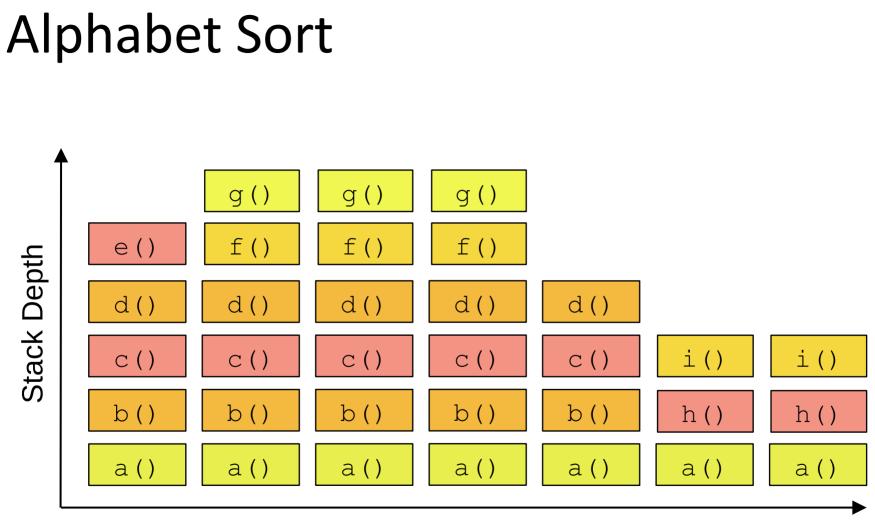
Stack Samples: Merged



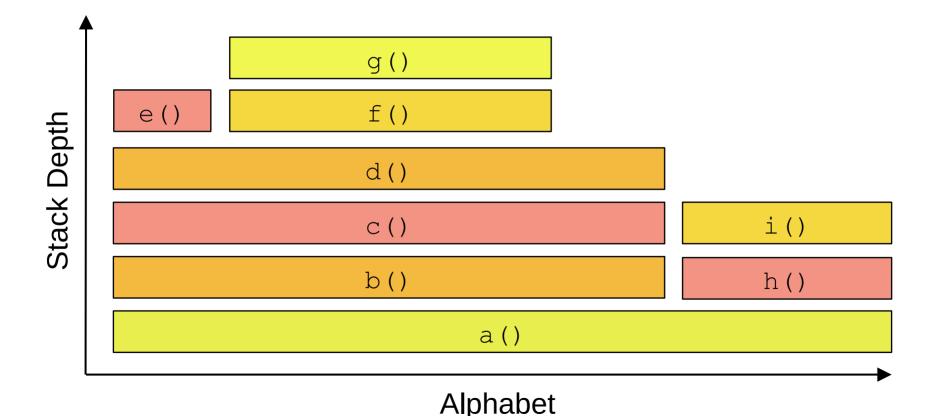
Example Profile: Merged

										111											
1000								1													
							1111	i i		111			1111								
			*****					1 1			1111111									****	
-		*****	*********				*****	1													
							n n n n n n														
	╶					n i The spir					*****										

															•••						
													-+			╡╫╸ _┙ ╴ _╋ ╴╋╓╴┍╖┇╶ _┲ ╷╝╖╴					
				· · · · · · · · · · · · · · · · · · ·	╏╌┲┸┲┸╼┥╓┥╴╴┸╴┙																
			Lic		Lio/ne	Lie	Lie				Lio/	Lio/ L	4			Lie/net Lie/n					-
L.	Li., Li.,	Lio/ne	Lio/	Lio			Lio		Lio/					Lio. L.		Lio/net Lio/n			LI		L.
Let .			Lio/	Lio.	Lio/ne		Lio		Lio/	and the second	Lio/	Lio/ L.		Lio/netty.		Lio/netty/chan.			Li.		L.,
L.	Li Li	Lio/netty	Lio/	Lio.	Lio/ne		Lio		Lio/		Lio/n	Lio/ L	-	Lio/netty.		Lio/netty/chan		Li	Li.	L···	L
L.	Li., Li.,	Lio/netty	Lio/	Lio	Lio/ne	Lio.	Lio.		Lio/	_	Lio/n	Lio/ L.		Lio/netty		Lio/netty/chan Lio/netty/chan		LI.	Li.	L.	L
T Let	Lio/net	Lio/netty	Lio/	Lio Int	Lio/ne	Lio/n	Lio		Lio/		Lio/n	Lio/net	-	Lio/netty.				Lio	In.	Lio/n	T
1	Interpr.	Interpreter	Inte	and the second se	Interp	Inter.	Int.		Inte	1	Inter.	Interpr.		Interpreter		Interpreter				Inter	1 T
1	Interpr.	Interpreter	Inte	Int	Interp	Inter.	Int.		Inte	1	Inter	Interpr.		Interpreter		Interpreter		Int.	In	Inter.	1
C		call_stub	call	cal	call_s	call	cal		call		call	call_stub		call_stub		call_stub JavaCalls::cal		cal	ca	call	1
J	JavaCal	JavaCalls JavaCalls	Java Java	Jav	JavaCa	JavaC	Jav		Java	-	JavaC JavaC	JavaCal JavaCal		JavaCalls.		JavaCalls::cal		Jav	Ja	JavaC.	1
J	JavaCal	JavaCalls	and the second se	Jav Jav	JavaCa	JavaC	Jav		Java	1		JavaCal.	_	JavaCalls		JavaCalls::cal		Jav	Ja	JavaC.	
5		A short water of the second	Java		JavaCa thread	JavaC	Jav		Java	J	JavaC.	and the second se		and the second				Jav	Ja th	JavaC threa	Contraction of the
t	thread		the second s	thr.		threa	thr.	_	thre	1		thread		thread_en	-	thread_entry(J.			Contraction of the		
	JavaThr.	JavaThrea	Java	Jav	JavaTh	JavaT.	Jav.		Java	5		JavaThr.		JavaThrea.		JavaThread::th.	and the second se	Jav.	Ja	JavaT	Contraction of the local distance of the loc
J.,	JavaThr.	JavaThrea	Java.	Jav	JavaTh	JavaT.	Jav	in	Java		0	JavaThr.		JavaThrea.		JavaThread::ru.		Jav	Ja	JavaT	-
j.,	java_st	java_star.	java	jav	java_s	java	jav		java	J	java	java_st	-	java_star		java_start(Thr.		a_sta	ja	java	1
S	start_t	start_thr	star	sta	start	start	sta	st	star.	5	start.	start_t		start_thr.		start_thread	sta	rt_th	St	start	9.1



Alphabet Merged ("Flame Graph")



Example Profile: Flame Graph

ja sta

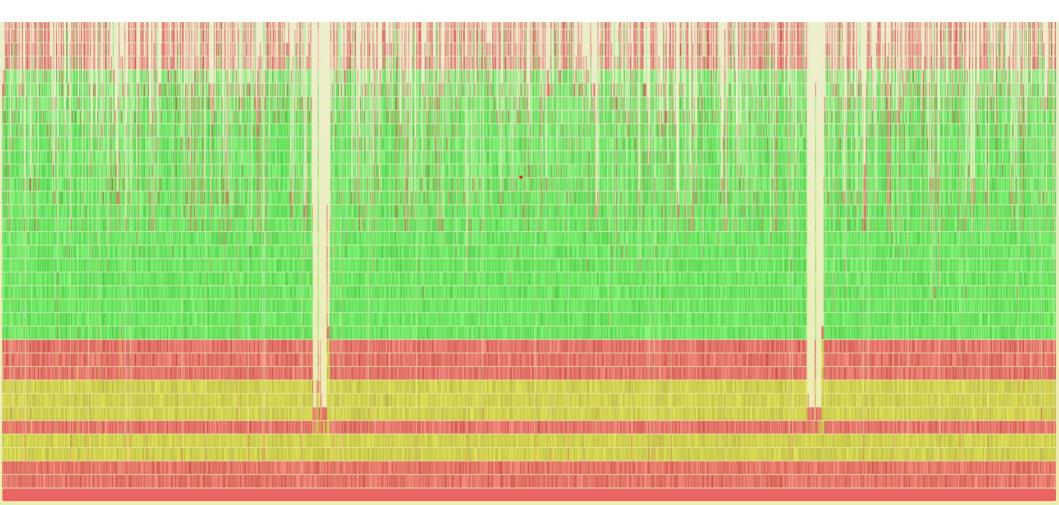
			sys_write	
	io/		system_call_fastpath	
	io/		write	
	org/		sun/nio/ch/FileDispatcherImpl.write0	
	io/ne		sun/nio/ch/SocketChannelImpl.write	
	io/ne		io/netty/buffer/PooledUnsafeDirectByteBuf.readB	1 (t.)
	org/mozilla/javas o o sun/refle		io/netty/channel/nio/AbstractNioByteChannel.doWrite	in
	org/mozilla/javasoror or. org/mozilla/javas	1	io/netty/channel/AbstractChannel\$AbstractUnsafe.f	do
	org/ org/mozilla/javas org/moz o or. org/mozill		io/netty/channel/DefaultChannelPipeline\$HeadConte.	SO
	org/mozilla/javascript/gen/filehome_bgregg_te org/mozilla	"	io/netty/channel/AbstractChannelHandlerContext.flush	SO
	org/mozilla/javascript/gen/file_home_bgregg_tes org/mozilla/		io/netty/channel/ChannelOutboundHandlerAdapter.flush	do
	org/mozilla/javascript/gen/filehome_bgregg_testtest_vert_x_2_1_4		io/netty/channel/AbstractChannelHandlerContext.flush	vfs.
	org/mozilla/javascript/gen/filehome_bgregg_testtest_vert_x_2_1_4		io/netty/channel/ChannelDuplexHandler.flush	sys
	org/vertx/java/core/http/impl/DefaultHttpServer\$ServerHandler.doMessageRe		io/netty/channel/AbstractChannelHandlerContext.flush	syst
	org/vertx/java/core/net/impl/VertxHandler.channelRead	io/	org/vertx/java/core/net/impl/VertxHandler.channelRe	read
	io/netty/channel/AbstractChannelHandlerContext.fireChannelRead	io/net	io/netty/channel/AbstractChannelHandlerContext.fire	sun/n
	io/netty/handler/codec/ByteToMessageDecoder.channelRead		io/netty/handler/codec/ByteToMessageDecoder.channelR	sun/ni
	io/netty/channel/AbstractChannelHandlerContext.fireChannelRead		io/netty/channel/AbstractChannelHandlerContext.fireC	io/net
	io/netty/channel/nio/AbstractNioByteChannel\$NioByteUnsafe.read			
	io/netty/channel/nio/NioEventLoop.processSelectedKey			
	io/netty/channel/nio/NioEventLoop.processSelectedKeysOptimized			
	lio/netty/channel/nio/NioEventLoop.processSelectedKeys			
	io/netty/channel/nio/NioEventLoop.run			
	Interpreter			
	Interpreter			
	call_stub			
	JavaCalls::call_helper			
	JavaCalls::call_virtual			
	JavaCalls::call_virtual			
	thread_entry			
	JavaThread::thread_main_inner			
	JavaThread::run			
	start			
-	thread			
va				

Example Profile: Flame Graph (with code hues)

St JavaThread::thread_main_inner GCT JavaThread::run java_start start_thread		I. I. I. I. I. Io/ Io/ Io/ <th>io/net</th> <th>sysem_call_fastpath system_call_fastpath write sun/nio/ch/FileDispatcherImpl.write0 sun/nio/ch/SocketChanneIImpl.write io/netty/buffer/PooledUnsafeDirectByteBuf.readB io/netty/channel/nio/AbstractNioByteChannel.doWrite io/netty/channel/AbstractChannelPipeline\$HeadConte io/netty/channel/DefaultChannelPipeline\$HeadConte io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/ChannelOutboundHandlerAdapter.flush io/netty/channel/ChannelDuplexHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.fire io/netty/channel/AbstractChannelHandlerContext.fire</th> <th>t in do so so do vfs sys syst read sun/n sun/ni io/net</th>	io/net	sysem_call_fastpath system_call_fastpath write sun/nio/ch/FileDispatcherImpl.write0 sun/nio/ch/SocketChanneIImpl.write io/netty/buffer/PooledUnsafeDirectByteBuf.readB io/netty/channel/nio/AbstractNioByteChannel.doWrite io/netty/channel/AbstractChannelPipeline\$HeadConte io/netty/channel/DefaultChannelPipeline\$HeadConte io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/ChannelOutboundHandlerAdapter.flush io/netty/channel/ChannelDuplexHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.flush io/netty/channel/AbstractChannelHandlerContext.fire io/netty/channel/AbstractChannelHandlerContext.fire	t in do so so do vfs sys syst read sun/n sun/ni io/net
GCT JavaThread::run		thread_entry			
java_startstartstartstartstartthread					
start_thread					
	-				
	start_f	hread			
Va la	va				

G ja st

Replay 1/3: Time Columns



Replay 2/3: Time Merged (aka "Flame Chart")

											a de sector a descriterá da				-						
																		2270 B721 B M			
	Ш																				
											Щ										
	Ш																				
	Ш																				
		Li Li	Lio/ne	Lio/	Lio	Lio/ne	Lio	Lio		Lio/	L	Lio/	Lio/ L	Lio L		Lio/net Lio/n			Li	L	L
	•	Li., Li.,	Lio/ne	Lio/	Lio	Lio/ne	Lio	Lio		Lio/	L	Lio/	Lio/ L	Lio/netty		Lio/netty/chan			Li	L	L
	•	Li., Li.,	Lio/netty	Lio/	Lio	Lio/ne	Lio	Lio		Lio/	L	Lio/n	Lio/ L	Lio/netty		Lio/netty/chan		Li	Li	L	L
	•	Li., Li.,	Lio/netty	Lio/	Lio	Lio/ne	Lio	Lio		Lio/		Lio/n	Lio/ L	Lio/netty		Lio/netty/chan		Li	Li	L	L
L		Lio/net	Lio/netty	Lio/	Lio	Lio/ne	Lio/n	Lio		Lio/		Lio/n	Lio/net	Lio/netty		Lio/netty/chan		Lio	Li	Lio/n	L
I.		Interpr	Interpreter	Inte	Int.	Interp	Inter	Int		Inte	A CONTRACTOR OF	Inter	Interpr.	Interpreter		Interpreter		Int	In		I
I.		Interpr	Interpreter	Inte	Int.	Interp	Inter.	Int.		Inte	Contraction of the local division of the loc	Inter.	Interpr.	Interpreter		Interpreter		Int	In	and the second se	I
C.		and the second sec	call_stub	call	cal	call_s	call	cal		call	Constanting of	call	call_stub	call_stub		call_stub		cal	ca	a second second and the second	с
ј.			JavaCalls	Java	Jav	JavaCa		Jav		Java	and the second second		JavaCal	JavaCalls		JavaCalls::cal		Jav	Service and the	JavaC	and the second second
J.			JavaCalls	Java	Jav	JavaCa	A second second second second	Jav		Java		and the second second second	JavaCal	JavaCalls		JavaCalls::cal		Jav	Ja	JavaC	Contraction of the
J.			JavaCalls	Java	Jav	JavaCa	JavaC	Jav		Java	1.		JavaCal	JavaCalls		JavaCalls::cal		Jav		JavaC	Second Street
t.		and the second	-	Contraction of the local division of the loc	thr.	thread	threa	thr.		thre	t		thread	thread_en		thread_entry(J			th	threa	
J.			JavaThrea		Jav	JavaTh	Reservation of the second	Jav		Java	J	and a second second second	JavaThr	JavaThrea		JavaThread::th		Jav	The March Street	JavaT	A CONTRACTOR OF
J.	-		JavaThrea	Java	Jav	JavaTh		Jav		Java	J.,		JavaThr	JavaThrea		JavaThread::ru		Jav		JavaT	
j .			java_star	java	jav		java	jav	and the second second	java	j	and the second second second second	java_st	java_star.		java_start(Thr	and the second second	_sta	A CONTRACTOR OF	java	j.,
S.	•	start_t	start_thr	star.	sta	start	start	sta	st	star	S.,	start	start_t	start_thr.		start_thread	star	t_th	st	start	S
Second Street																					A CONTRACTOR OF THE OWNER

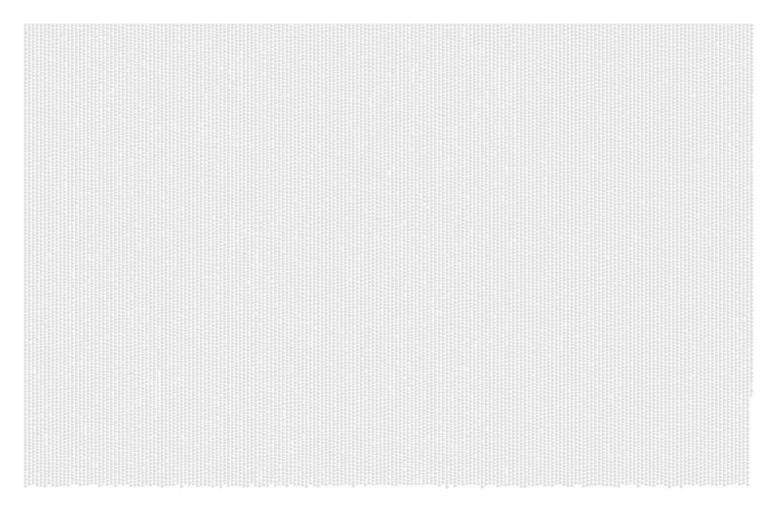
Replay 3/3: Flame Graph

			sys_write	
	io.		system_call_fastpath	
	io/		write	111
	org		sun/nio/ch/FileDispatcherImpl:.write0	i ii i
	io/n		sun/nio/ch/SocketChannelImpl:.write	
	o org/m o io/n		io/netty/buffer/PooledUnsafeDirectByteBuf:.read	t
	org/mozilla/javas		io/netty/channel/nio/AbstractNioByteChannel:.doWr	in
	org/mozilla/javasc o or or org/mozil		io/netty/channel/AbstractChannel\$AbstractUnsafe:	do
	org org/mozilla/javasc org/mo o org org/mozil	1	io/netty/channel/DefaultChannelPipeline\$HeadContex	soc
	org/mozilla/javascript/gen/file_home_bgregg_te org/mozill		io/netty/channel/AbstractChannelHandlerContext:.fl	soc
	org/mozilla/javascript/gen/filehome_bgregg_tes org/mozilla/		io/netty/channel/ChannelOutboundHandlerAdapter:.fl	do
	org/mozilla/javascript/gen/filehome_bgregg_testtest_vert_x_2_1		io/netty/channel/AbstractChannelHandlerContext:.fl	vfs
	org/mozilla/javascript/gen/filehome_bgregg_testtest_vert_x_2_1_4		io/netty/channel/ChannelDuplexHandler:.flush	sys
	org/vertx/java/core/http/impl/DefaultHttpServer\$ServerHandler:.doMessage		io/netty/channel/AbstractChannelHandlerContext:.flush	syste
	org/vertx/java/core/net/impl/VertxHandler:.channelRead	io/	org/vertx/java/core/net/impl/VertxHandler:.channelR	read
i	p/netty/channel/AbstractChannelHandlerContext:.fireChannelRead	io/nett	io/netty/channel/AbstractChannelHandlerContext:.fir	sun/ni
io	(netty/handler/codec/ByteToMessageDecoder:.channelRead		io/netty/handler/codec/ByteToMessageDecoder:.channe	sun/ni
io,	netty/channel/AbstractChannelHandlerContext:.fireChannelRead		io/netty/channel/AbstractChannelHandlerContext:.fire	io/net
io/n	etty/channel/nio/AbstractNioByteChannel\$NioByteUnsafe:.read			
io/ne	tty/channel/nio/NioEventLoop:.processSelectedKey			
io/ne	ty/channel/nio/NioEventLoop:.processSelectedKeysOptimized			
io/ne	ty/channel/nio/NioEventLoop:.processSelectedKeys			
io/ne	:ty/channel/nio/NioEventLoop:.run			
Inter	preter			
Inter	preter			
call_s	tub			
Java	alls::call_helper			
Java	alls::call_virtual			
Java	alls::call_virtual			
	d_entry			
St., JavaT	hread::thread_main_inner			
GC JavaT	hread::run			
java_start				
start_threa	d			
invo				

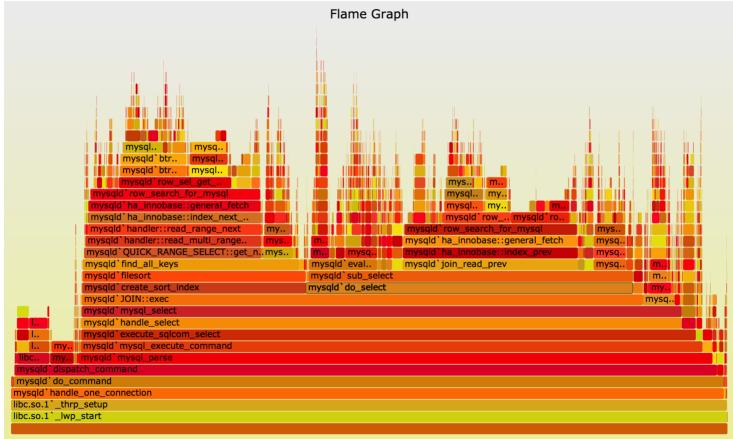
Origin (2011): CPU Profiling

```
# dtrace -x ustackframes=100 -n 'profile-997 /execname == "mvsgld"/ {
    @[ustack()] = count(); \} tick-60s { exit(0); }'
[... over 500,000 lines truncated ...]
              libc.so.1` priocntlset+0xa
              libc.so.1`getparam+0x83
              libc.so.1`pthread getschedparam+0x3c
              libc.so.1`pthread setschedprio+0x1f
              mysqld` Z16dispatch command19enum server commandP3THDPcj+0x9ab
              mysqld` Z10do commandP3THD+0x198
              mysqld`handle one connection+0x1a6
              libc.so.1` thrp setup+0x8d
              libc.so.1` lwp start
             4884
              mysqld` Z13add to statusP17system status varS0 +0x47
              mysqld` Z22calc sum of all statusP17system status var+0x67
              mysqld` Z16dispatch command19enum server commandP3THDPcj+0x1222
              mysqld` Z10do commandP3THD+0x198
              mysqld`handle one connection+0x1a6
              libc.so.1` thrp setup+0x8d
              libc.so.1` lwp start
             5530
```

Full output



... as a Flame Graph



Function: mysqld`do_select (159,007 samples, 45.64%)

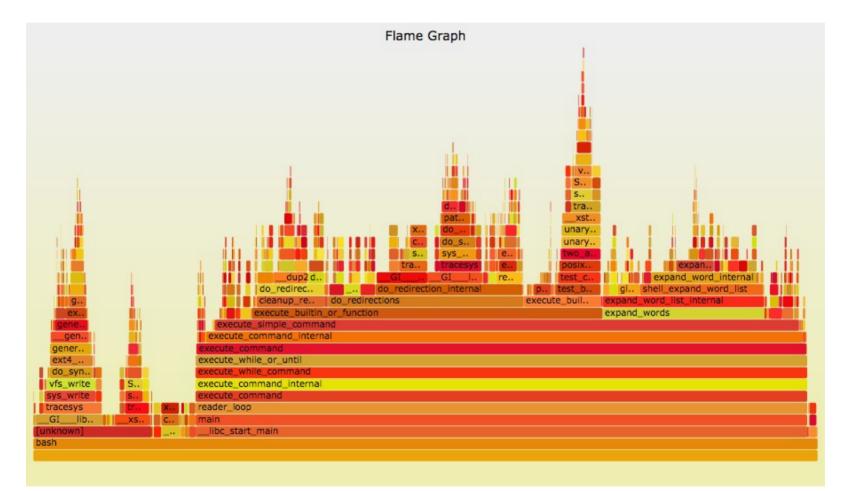
Linux example: perf Profiling

```
# perf record -F 99 -ag -- sleep 30
[ perf record: Woken up 9 times to write data ]
[perf record: Captured and wrote 2.745 MB perf.data (~119930 samples) ]
# perf report -n -stdio
[...]
# Overhead Samples Command Shared Object
                                                                        Symbol
  20.42%
                 605
                         bash [kernel.kallsyms] [k] xen hypercall xen version
              --- xen hypercall xen version
                                                                              call tree
                 check events
                                                                              summary
                 |--44.13%-- syscall trace enter
                           tracesvs
                           |--35.58%-- GI libc fcntl
                                      |--65.26%-- do redirection internal
                                               do redirections
                                               execute builtin or function
                                               execute simple command
[... ~13,000 lines truncated ...]
```

Full perf Output

	Note Note
--	--

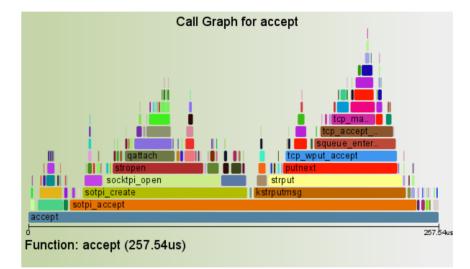
... as a Flame Graph



Inspiration

Neelakanth Nadgir's function_call_graph.rb (2007):

- It was inspired by Roch Bourbonnais's CallStackAnalyzer, which was inspired by Jan Boerhout's vftrace.
- The x-axis is time, and it shows a complete function trace.
- Flame graphs are different: The x-axis is the *population*, and they can show function traces or *stack samples*.



more flamegraph.pl
[...]
This was inspired by Neelakanth Nadgir's excellent function call graph.rb
program, which visualized function entry and return trace events. As Neel
wrote: "The output displayed is inspired by Roch's CallStackAnalyzer which
was in turn inspired by the work on vftrace by Jan Boerhout". See:
https://blogs.oracle.com/realneel/entry/visualizing_callstacks_via_dtrace_and
[...]

Image source: https://blogs.oracle.com/realneel/entry/visualizing_callstacks_via_dtrace_and

Flame Graph Summary

Visualizes a collection of stack traces

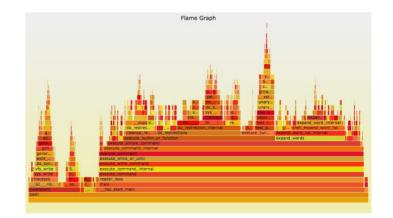
- x-axis: population: e.g., alphabetical sort to maximize merging
- y-axis: stack depth
- **color**: random (default) or a dimension

Original implementation: Perl + SVG + JavaScript

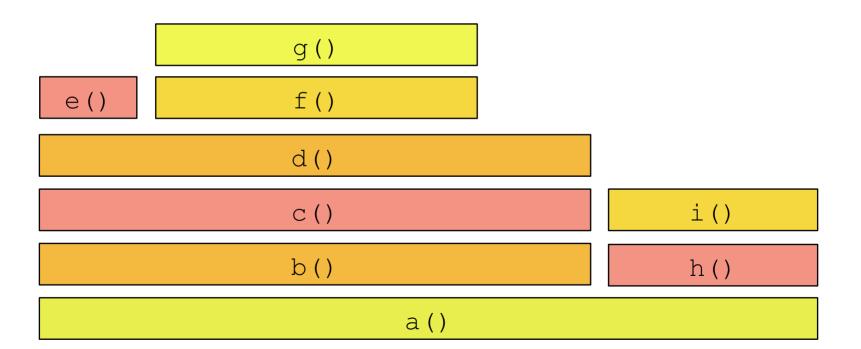
- https://github.com/brendangregg/FlameGraph
- Takes input from many different profilers

References:

- http://www.brendangregg.com/flamegraphs.html
- http://queue.acm.org/detail.cfm?id=2927301
- "The Flame Graph" CACM, June 2016

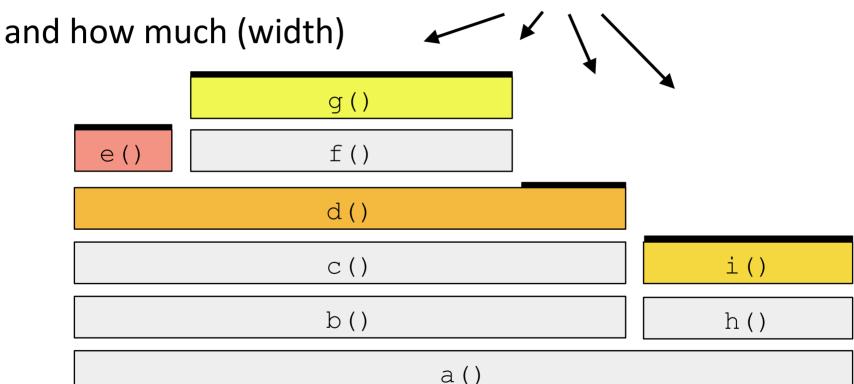


Flame Graph Interpretation



Flame Graph Interpretation (1/4)

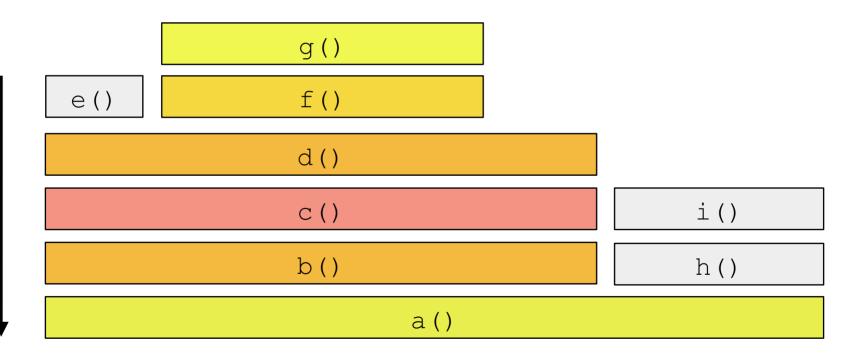
Top edge shows who is running on-CPU,



Flame Graph Interpretation (2/4)

Top-down shows ancestry

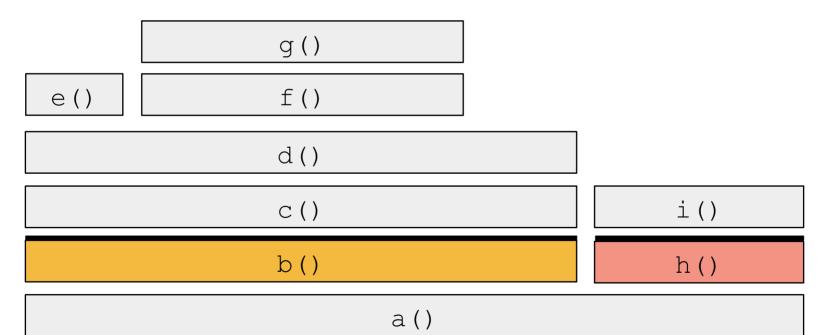
e.g., from g():



Flame Graph Interpretation (3/4)

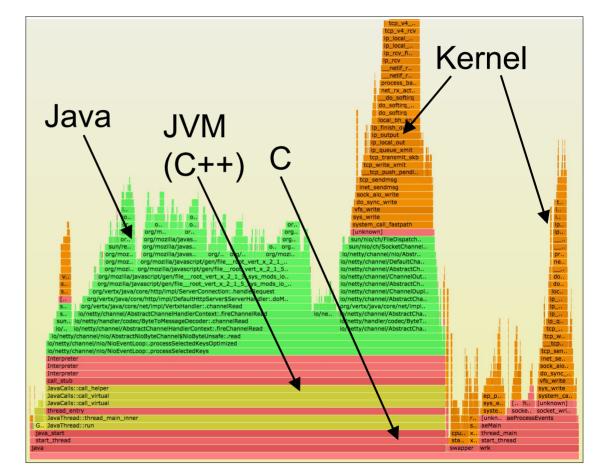
Widths are proportional to presence in samples

e.g., comparing b() to h() (incl. children)



Flame Graph Interpretation (4/4)

Colors randomized to differentiate frames Or used for code type; e.g.: green == JIT (e.g., Java) aqua == inlined red == user-level orange == kernel yellow == C++ magenta == search term



CPU Flame Graph Tips & Tricks

A) Check sample count (bottom frame): idle system?

E.g., 49 Hertz x 30 sec x 16 CPUs == 23,520 samples at 100% CPU utilization.
 <500 samples total would mean <2% busy and probably not interesting!

B) Off-CPU time (I/O, locks) not present

- But their initialization/spin breadcrumbs may be present
- Can use off-CPU flame graphs for this (covered later)

C) Some tiny CPU code paths may be missing

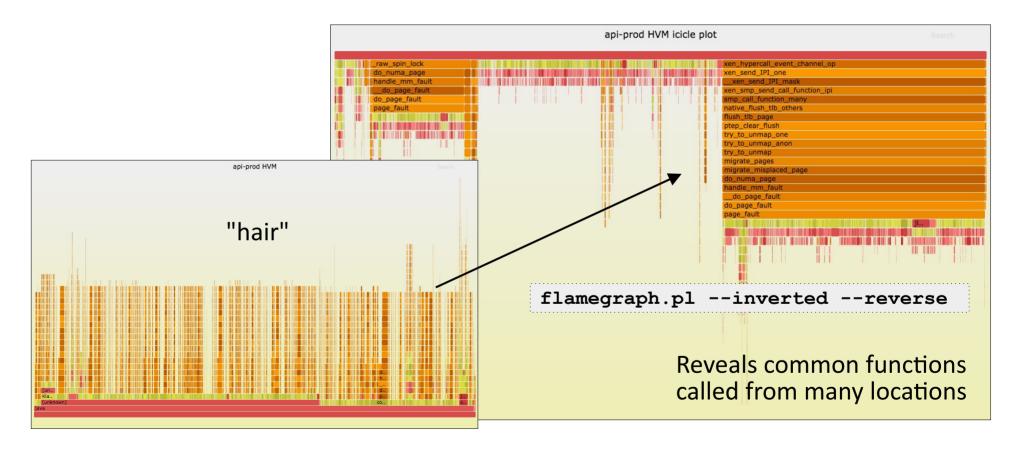
- E.g., some kernel code paths that disable interrupts, and hypervisor time from the guest.
- This is a detail of the profiler and target. Flame graphs just show what the profiler sees.
- Flame graphs may drop tiny frames that are <1 pixel wide unless zoomed in, just to speed up rendering.

D) Too much "hair"? Try a leaf merge...

Merge stack frames from leaf to root. The default is root to leaf.



Icicle Graph with Leaf Merge



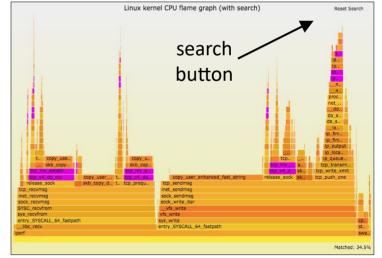
Flame Graph Interactivity

Essentials:

- Mouse-over for frame info (tool tips, status bar)
- Click to zoom
- Search (Ctrl-F or button)

Nice to have:

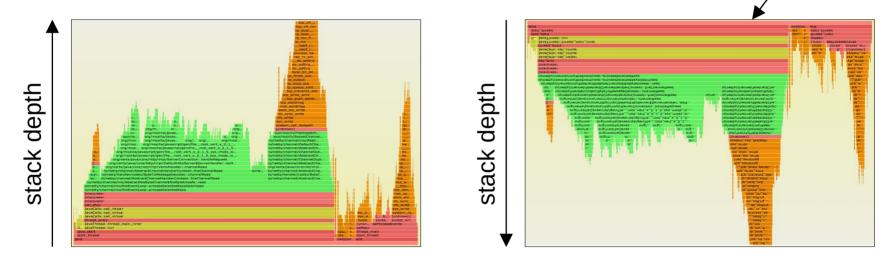
- Merge control: root, leaf, middle
- Y-axis direction: flame or icicle
- Flame chart toggle
- Canned searches
- Collapse filters
- Code links



search matches in magenta

Which way up?

My original flamegraph.pl has --inverted for an "icicle graph"



Either way is fine!

- Icicle layout helps avoid scrolling when starting at the top
- Let the end-user choose

Current View: Flame Graph
V
Reset Zoon Flip Flame Graph
iisexpress
DomainBoundILStubClass.IL_STUB_COMtoCLR(lo
BookService.WebApiApplication.Application_St
Source: https://learn.microsoft.com/en-us/visualstudio/profiling/flame-graph

Differential Flame Graphs

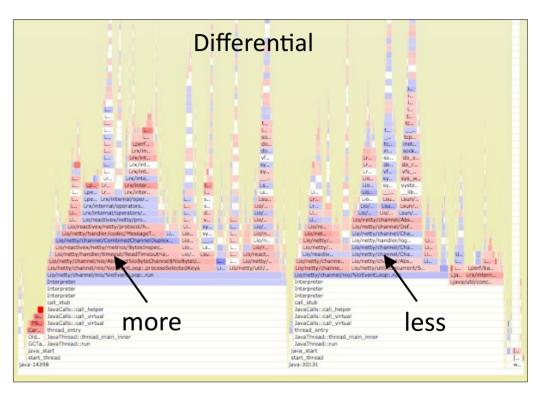
Hues:

- red == more samples
- blue == less samples

Intensity:

- Degree of difference
- Other examples
 - flamegraphdiff

This spectrum can show other metrics, like CPI. Remember to show elided frames!



Poor Man's Differential Flame Graphs

Toggle between tabs in your browser

- Like searching for Pluto!

Or flip between slides

- Cue exciting demo!



Poor Man's Differential Flame Graphs

Toggle between tabs in your browser

- Like searching for Pluto!

Or flip between slides

- Cue exciting demo!



System CPU Profilers

- Linux
 - perf_events (aka "perf")
 - bcc profile (eBPF-based)
- Windows
 - XPerf, WPA (now has flame graphs!)
- OS X
 - Instruments
- And many others...

Tip: use system profilers whenever possible. Runtime profilers (e.g., Java JVMTI-based) are user space and typically don't include kernel CPU time or kernel stacks.

Linux CPU Flame Graphs

Linux 5.8+ via perf for *simplicity* (2020):

```
perf script flamegraph -F 49 -a -- sleep 30
```

– Generates flamegraph.html. One command! Thanks Andreas Gerstmayr.

Linux 4.9+ via eBPF for *efficiency* (2016):

```
apt-get install bpfcc-tools
git clone https://github.com/brendangregg/FlameGraph
profile-bcc.py -dF 49 30 | ./FlameGraph/flamegraph.pl > perf.svg
```

- eBPF (no longer an acronym) is the name of an in-kernel execution environment, used in this case for aggregating stack samples in kernel context
- Most efficient: no perf.data file, summarizes in-kernel

Some runtimes (e.g., JVM) require extra steps for stacks & symbols (next section)

Older Linux CPU Flame Graphs

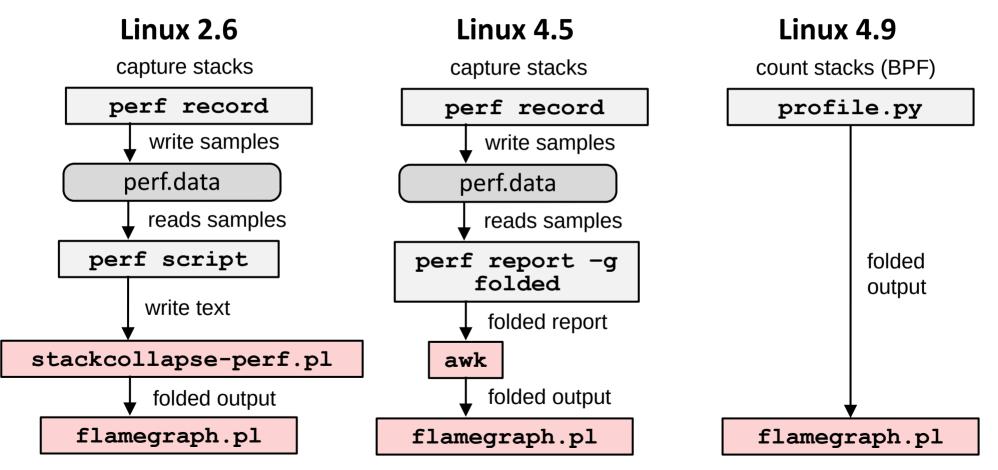
Linux 2.6+ via perf.data and perf script (2009):

git clone https://github.com/brendangregg/FlameGraph; cd FlameGraph
perf record -F 49 -a -g -- sleep 30
perf script | ./stackcollapse-perf.pl |./flamegraph.pl > perf.svg

Linux 4.5 can use folded output (2016):

 Skips the CPU-costly stackcollapse-perf.pl step; see: http://www.brendangregg.com/blog/2016-04-30/linux-perf-folded.html

Linux Profiling Optimizations



GUI Automation

There are many options nowadays. I've worked on five:

- **Netflix Vector** (now retired!):



- Netflix FlameScope (covered later)
- **Netflix FlameCommander** (continuous profiling; not open source yet)
- I'm now helping with Intel vTune and Intel gProfiler

Open source examples include Granulate gProfiler, Eclipse TraceCompass, Grafana flame graphs, Firefox profiler, and more (see implementation slides). Build your own!

Flame Charts (2013)

Inspired by flame graphs: https://bugs.webkit.org/show_bug.cgi?id=111162

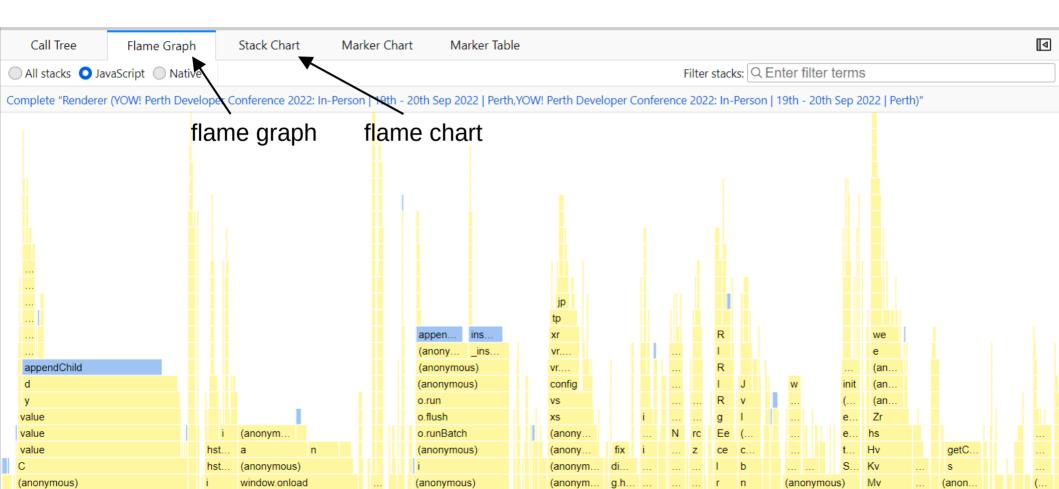
Ilya Tikhonovsky	2013-03-01 04:33:26 PST		Description
Flame Chart may gi	ve to the developer a better clue what is going on wit	n the performance without expanding the	entire tree.
<u>http://dtrace.org/</u>	<u>blogs/brendan/2011/12/16/flame-graphs/</u>		
Ilya Tikhonovsky	2013-03-01 04:35:00 PST		Comment 1
Created <u>attachment</u> screenshot	<u> 190933 [details]</u>	0.11% 0.11% est interfTML 0.11% 0.21% > 5z	is7name=ful5am_uthuse=0.1008 。
Ilya Tikhonovsky	2013-03-01 04:37:34 PST		
Created <u>attachment</u> Patch	<u> 190934 [details]</u>		

Chrome DevTools Flame Charts (2022)

	Elements	Console Sou	irces Networ	k Performance	e Memory	Application	Security	Lighthouse	Recorder 👗	Performance	insights 👗	
• •		skillsmatter.co	om #1 🔹 🔻	Screenshots	Memory	Web Vitals	T					
	1000 ms	2000 ms	3000 m	s 400	0 ms	5000 ms	6000 r	70	00 ns	8000 ms	9000 ms	
	ork ttern-test-d7c7				ms 6300 ms t (analytics.twitte		6400 ms linkid.js (v		500 ms 6550 n collect p.gi		6650 ms	6700 ms
▼ Main T	— https://skillsmat	ter.com/conferences Task Tid FuII Mv Kv Hv hs Zr ((e	/13/32-yow-pertr		Task Evalrip	Task t Eveded						

Total blocking time: 2152.88ms (estimated) Learn more

Firefox Profiler Flame Graph (2022)



Flame Charts x-axis: time

Flame Graphs x-axis: population

alphabet sort or another frame merging algorithm

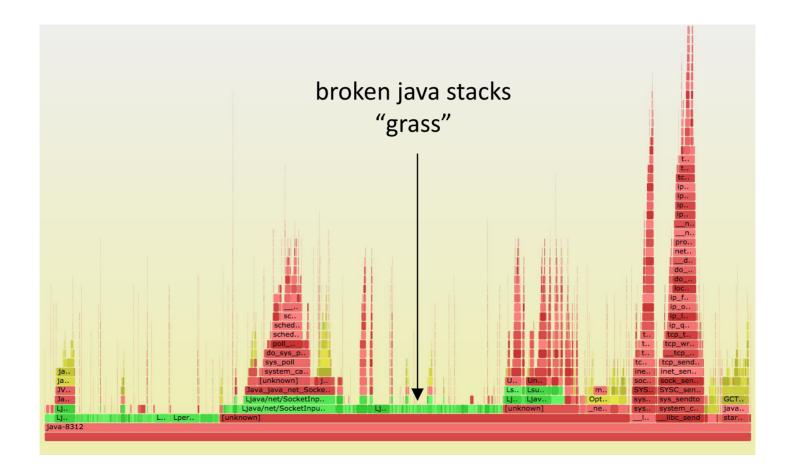
3. STACKS AND SYMBOLS

And Other Issues

Broken Stack Traces are Common

```
# perf record -F 99 -a -g - sleep 30
# perf script
[...]
java 4579 cpu-clock:
            7f417908c10b [unknown] (/tmp/perf-4458.map)
java 4579 cpu-clock:
            7f41792fc65f [unknown] (/tmp/perf-4458.map)
            a2d53351ff7da603 [unknown] ([unknown])
[...]
            should probably have more frames
```

... as a Flame Graph



Fixing Stack Walking

A. Frame pointer-based

- Fix by disabling that compiler optimization: gcc's -fno-omit-frame-pointer
- Pros: simple, supported by many tools
- Cons: might cost a little extra CPU (usually <1%)

B. Debug info (DWARF) walking

Cons: costs disk space, and not supported by all profilers, expensive for real-time tracing

C. JIT-provided runtime walkers

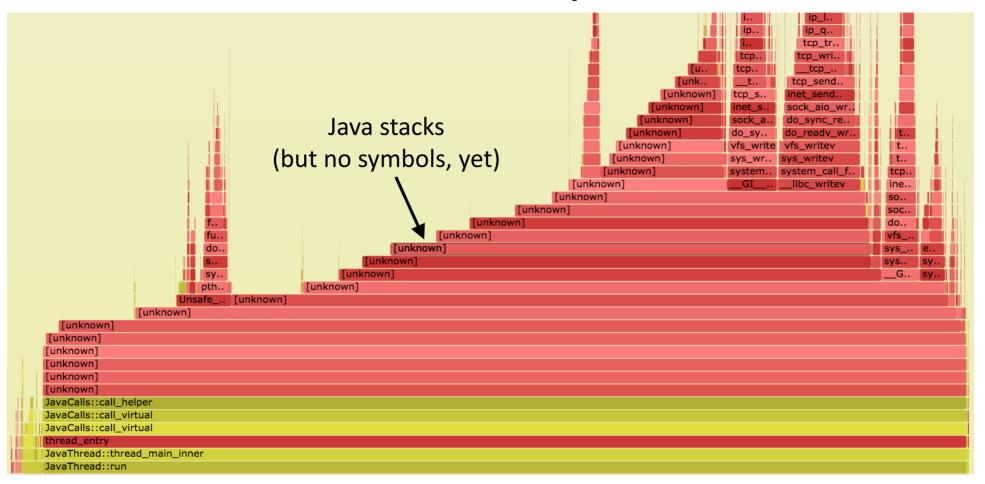
- Pros: include more internals, such as inlined frames (e.g., JVMTI stacks)
- Cons: limited to application internals: no kernel
- D. Last branch record (LBR)
- E. Add-on walkers (eBPF)

Fixing Java Stack Traces

I prototyped JVM frame pointers. Oracle rewrote it and included it in Java as -XX:+PreserveFramePointer (JDK 8 u60b19)

4	anf comint	
	perf script	
[] 		
Jav	<pre>ra 8131 cpu-clc 7fff76f2dce1</pre>	
		os::javaTimeMillis() (/usr/lib/jvm
	7fd301861e46	[unknown] (/tmp/perf-8131.map)
	7fd30184def8	[unknown] (/tmp/perf-8131.map)
	7fd30174f544	[unknown] (/tmp/perf-8131.map)
	7fd30175d3a8	[unknown] (/tmp/perf-8131.map)
	7fd30166d51c	[unknown] (/tmp/perf-8131.map)
	7fd301750f34	[unknown] (/tmp/perf-8131.map)
	7fd3016c2280	[unknown] (/tmp/perf-8131.map)
	7fd301b02ec0	[unknown] (/tmp/perf-8131.map)
	7fd3016f9888	[unknown] (/tmp/perf-8131.map)
	7fd3016ece04	[unknown] (/tmp/perf-8131.map)
	7fd30177783c	[unknown] (/tmp/perf-8131.map)
	7fd301600aa8	[unknown] (/tmp/perf-8131.map)
	7fd301a4484c	[unknown] (/tmp/perf-8131.map)
	7fd3010072e0	[unknown] (/tmp/perf-8131.map)
	7fd301007325	[unknown] (/tmp/perf-8131.map)
	7fd301007325	[unknown] (/tmp/perf-8131.map)
	7fd3010004e7	[unknown] (/tmp/perf-8131.map)
	7fd3171df76a	JavaCalls::call_helper(JavaValue*,
	7fd3171dce44	JavaCalls::call_virtual(JavaValue*
	7fd3171dd43a	JavaCalls::call_virtual(JavaValue*
	7fd31721b6ce	<pre>thread_entry(JavaThread*, Thread*)</pre>
	7fd3175389e0	<pre>JavaThread::thread_main_inner() (/</pre>
	7fd317538cb2	JavaThread::run() (/usr/lib/jvm/nf
	7fd3173f6f52	java_start(Thread*) (/usr/lib/jvm/
	7fd317a7e182	start_thread (/lib/x86_64-linux-gn
		——————————————————————————————————————

Fixed Stacks Flame Graph



Fixing Native Symbols

- A. Add a -dbgsym package, if available
- B. Recompile from source

Fixing JIT Symbols (Java, Node.js, ...)

Just-in-time runtimes don't have a pre-compiled symbol table

So Linux perf looks for an externally provided symbol file

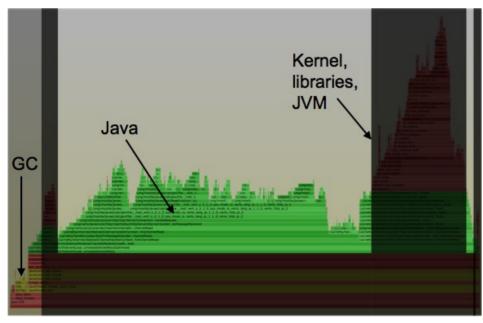
[...]

- This can be created by runtimes; e.g., Java's perf-map-agent
- Not the only solution; can also integrate with JIT-based walkers, or have an external symbol translator (perf script, or eBPF-based).

Fixed Symbols (zoom)

	tcp_transmit_skb				
	tcp_write_xmit				
tcp_push_pending_frames					
	tcp_sendmsg				
	inet_sendmsg				
	sock_aio_write				
	do_sync_write				
	vfs_write				
	sys_write				
	system_call_fastpath				
	[unknown]				
	Lsun/nio/ch/FileDispatcherImpl:.write0				
	Lsun/nio/ch/SocketChannelImpl:.write				
	Lio/netty/channel/nio/AbstractNioByteChannel:.doWrite				
	Lio/netty/channel/DefaultChannelPipeline\$HeadContext:.flush				
	Lio/netty/channel/AbstractChannelHandlerContext:.flush				
	Lio/netty/channel/ChannelOutboundHandlerAdapter:.flush				
	Lio/netty/channel/AbstractChannelHandlerContext:.flush				
	Lio/netty/channel/ChannelDuplexHandler:.flush				
	Lio/netty/channel/AbstractChannelHandlerContext:.flush				
Lio/	Lorg/vertx/java/core/net/impl/VertxHandler:.channelReadComplete				

2014: Java Profiling (broken stacks)

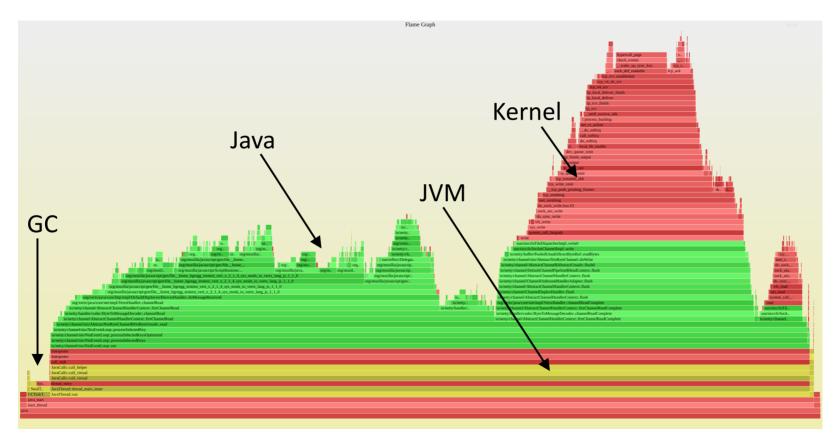


Java JVM JVM JVM State S

System Profilers

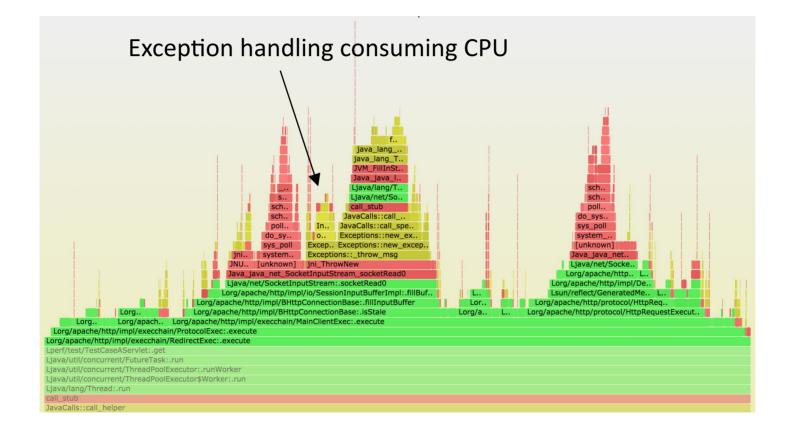
Java Profilers

2018: Java Profiling (fixed stacks)



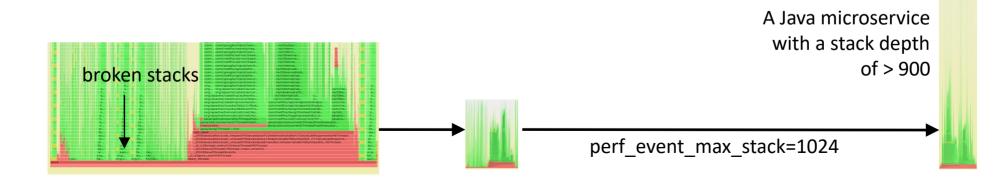
CPU Mixed-mode Flame Graph

Mixed-Mode Case Study



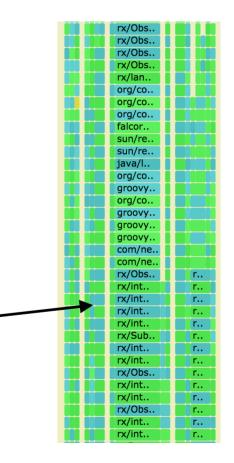
Other Issues

- JIT Symbol churn
 - Take before and after snapshots, or use perf's timestamped symbol logs.
- Containers
 - Are symbol files read from the right namespace? Should now work.
- Stack Depth limits
 - Linux perf had a 127 frame limit, now tunable. Thanks Arnaldo Carvalho de Melo!



Inlining

- Many frames may be missing (inlined)
 - Flame graph may still make enough sense
- Inlining can often be be tuned
 - e.g. Java's -XX:-Inline to disable, but can be 80% slower
 - Java's -XX:MaxInlineSize and -XX:InlineSmallCode can be tuned a little to reveal more frames: can even improve performance!
- Runtimes can un-inline on demand
 - So that exception stack traces make sense
 - e.g. Java's perf-map-agent can un-inline (unfoldall option)



Language/Runtime Issues

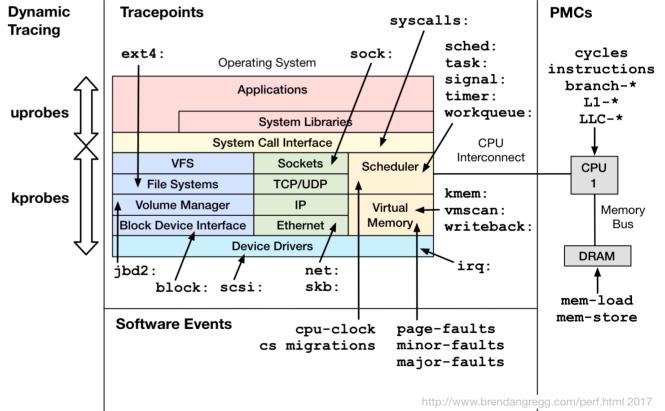
Each may have special stack/symbol instructions

- Java, Node.js, Python, Ruby, C++, Go, ...
- See: https://www.brendangregg.com/FlameGraphs/cpuflamegraphs.html
- Check if flame graphs are already in the "official" profiler
- Try an Internet search

4. ADVANCED FLAME GRAPHS

Flame graphs can visualize any stack trace collection

On Linux, stacks from any of these events:

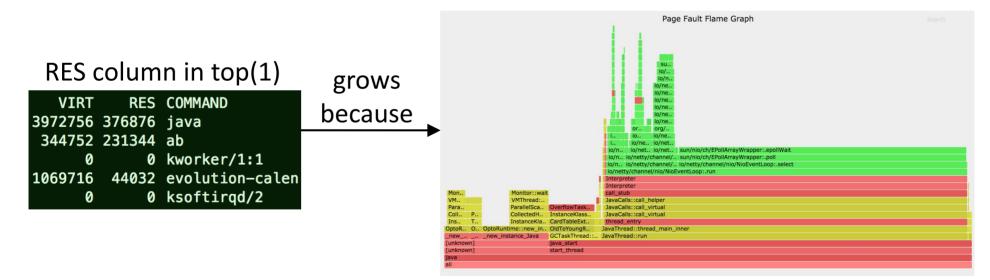


Page Faults

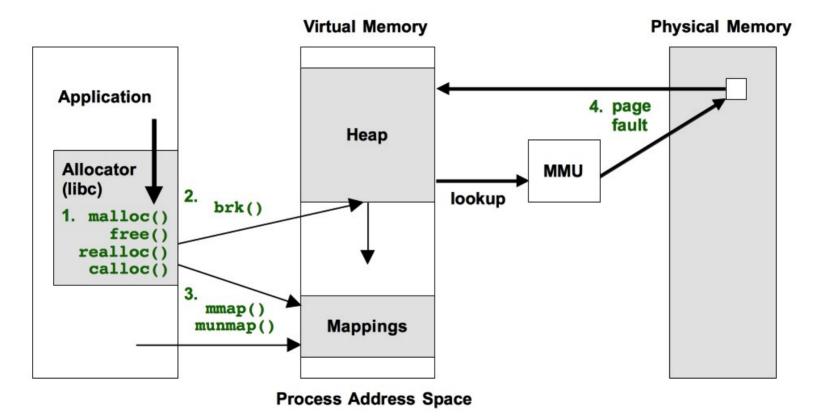
Show what triggered main memory (resident) to grow:

perf record -e page-faults -p PID -g -- sleep 120

- "fault" as (physical) main memory is allocated on-demand, when a virtual page is first populated
- Low overhead tool to solve some types of memory leak



Other Memory Sources



http://www.brendangregg.com/FlameGraphs/memoryflamegraphs.html

Disk I/O Requests

Shows who issued disk I/O (sync reads & writes):

perf record -e block:block_rq_insert -a -g -- sleep 60

GC? This JVM has swapped out!	Reset Zoom	Block I/O F	lame Graph	
GC? This JVM has swapped out! bik_finish_plug do_page_cache_readahead do_page_fault elv_add_request do_page_fault bk, finish_plug do_page_fault bk, finish_plug bk finish_plug do_page_cache_readahead bk_finish_plug do_page_cache_readahead bk_finish_plug do_page_cache_readahead bk_finish_plug do_page_fault do_page_fault do_fault do_page_fault do_fault do_page_fault do_fault do_page_fault do_page_fault				
GC? This JVM has swapped out!				
GC? This JVM has swapped out! [lemap_fault do_fault handle_mm_fault do_page_fault filemap_fault do_page_fault do_page_fault do_page_fault filemap_fault do_page_fault filemap_fault do_page_fault filemap_fault do_page_fault filemap_fault do_page_fault filemap_fault fi				
elv_add_request do_fault elv_add_request do_page_fault blk_flush_plug_list elv_add_request do_page_fault do_page_fault do_page_cache_readahead blk_flush_plug_list page_fault do_page_cache_readahead blk_flush_plug_list page_fault do_fault do_page_fault do_page_fault do_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_caclls::call_virtual do_page_fault doverflowTaskQueeus_ javaCalls::call_virtual do_page_fault doverflowTaskQueeus_ javaThread::thread_main_inner doctaskThread::run doverflowTask::do_it javaThread::thread_main_inner				
elv_add_request do_fault elv_add_request do_page_fault blk_flush_plug_list elv_add_request do_page_fault do_page_fault do_page_cache_readahead blk_flush_plug_list page_fault do_page_cache_readahead blk_flush_plug_list page_fault do_fault do_page_fault do_fault do_fault do_page_fault do_fault do_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_	GC? Th	his JVM has swapped out!		
Image: startImage: startelv_add_requestdo_page_faultblk_flush_plug_listelv_add_requestdo_page_faultdo_page_faultdo_page_cache_readaheadblk_flush_plug_listdo_page_cache_readaheadblk_finish_plugdo_page_cache_readaheadblk_finish_plugdo_faultdo_page_faultdo_faultdo_page_faultdo_faultdo_page_faultdo_faultdo_page_fault <td< td=""><td></td><td></td><td></td><td></td></td<>				
elv_add_requestdo_page_faultblk_flush_plug_listelv_add_requestdo_page_faultblk_fnish_plugblk_flush_plug_listpage_faultdo_page_cache_readaheadblk_finish_plugJVM_MonitorWaitdo_page_cache_readaheadswapin_readaheadInterpreterfilemap_faultdo_page_faultInterpreterdo_faultdo_page_faultInterpreterdo_faultdo_page_faultInterpreterdo_page_faultdo_page_faultJavaCalls::call_helperdo_page_faultdo_page_faultJavaCalls::call_virtualdo_page_fault				
blk_flush_plug_listelv_add_requestdo_page_faultblk_finish_plugblk_flush_plug_listpage_faultdo_page_cache_readaheadblk_finish_plugJVM_MonitorWaitra_submitswapin_readaheadInterpreterfilemap_faulthandle_mm_faultInterpreterdo_faultdo_page_faultInterpreterhandle_mm_faultdo_page_faultInterpreterdo_page_faultdo_page_faultdo_page_faultdo_page_faultdo_page_faultdo_page_faultdo_page_faultdo_page_faultdo_calls::call_virtualdo_page_faultdo_page_faultdo_calls::call_virtualdo_page_faultdo_talls::call_virtualdo_talls::call_virtualdo_page_faultdo_talls::call_virtualdo_talls::call_virtualdo_page_faultdo_talls::call_virtualdo_talls::call_virtualdo_page_faultdo_talls::call_virtualdo_talls::call_virtualdo_page_faultdo_talls::call_virtualdo_talls::call_virtualdo_page_faultdo_talls::call_virtualdo_talls::call_virtualdo_page_faultdo_talls::call_virtualdo_talls::call_virtualdo_page_faultdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_virtualdo_talls::call_vi				
blk_finish_plug blk_flush_plug_list page_fault do_page_cache_readahead blk_finish_plug JVM_MonitorWait ra_submit swapin_readahead Interpreter filemap_fault handle_mm_fault Interpreter do_page_fault do_page_fault Interpreter handle_mm_fault do_page_fault call_stub do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault do_page_fault avaCalls::call_wirtual page_fault overflowTaskQueueS avaCalls::call_virtual page_fault overflowTaskQueueS avaCalls::call_virtual page_fault overflowTaskQueueS avaCalls::call_virtual ParCompactionManager::push_objarray	elv_add_request			
do_page_cache_readaheadblk_finish_plugJVM_MonitorWaitra_submitswapin_readaheadInterpreterfilemap_faulthandle_mm_faultInterpreterdo_faultdo_page_faultInterpreterhandle_mm_faultdo_page_faultcall_stubdo_page_faultdo_page_faultJavaCalls::call_helperdo_page_faultdoverflowTaskQueueSJavaCalls::call_virtualpage_faultdoverflowTaskQueueSJavaCalls::call_virtualParCompactionManager::push_objarraydatabeet CardTableExtension:thread_entryMarkFromRootsTask::do_itdoverflowTagkQueuteSJavaThread::thread_main_innerjava_start				
ra_submitswapin_readaheadInterpreterfilemap_faulthandle_mm_faultInterpreterdo_faultdo_page_faultInterpreterhandle_mm_faultdo_page_faultInterpreterhandle_mm_faultdo_page_faultJavaCalls::call_helperdo_page_faultJavaCalls::call_wirtualJavaCalls::call_virtualpage_faultInstanceKlass::oopJavaCalls::call_virtualpage_faultInstanceKlass::oopJavaCalls::call_wirtualParCompactionManager::push_objarrayCardTableExtension:thread_entryMarkFromRootsTask::do_itOldToYoungRootsTas.JavaThread::thread_main_innerjava_startStart_threadJavaThread::run				
filemap_fault handle_mm_fault Interpreter do_fault do_page_fault Interpreter handle_mm_fault do_page_fault Interpreter handle_mm_fault do_page_fault Call_stub do_page_fault gage_fault JavaCalls::call_helper do_page_fault OverflowTaskQueueS. JavaCalls::call_virtual page_fault InstanceKlass::oop JavaCalls::call_virtual ParCompactionManager::push_objarray CardTableExtension: thread_entry MarkFromRootsTask::do_it OldToYoungRootsTas JavaThread::thread_main_inner GCTaskThread::run JavaThread::run start_thread	do_page_cache_readahead		blk_finish_plug	JVM_MonitorWait
do_faultInterpreterhandle_mm_faultdo_page_faultcall_stubdo_page_faultjavaCalls::call_helperdo_page_faultjavaCalls::call_virtualdo_page_faultjavaCalls::call_virtualpage_faultjavaCalls::call_virtualpage_faultjavaCalls::call_virtualpage_faultjavaCalls::call_virtualpage_faultjavaCalls::call_virtualpage_faultjavaCalls::call_virtualpage_faultjavaCalls::call_virtualpage_faultjavaCalls::call_virtualpage_faultjavaCalls::call_virtualpage_faultjavaThread::runGCTaskThread::runjavaThread::thread_main_innerjava_startjavaThread::runstart_threadjavaThread::run	ra_submit		swapin_readahead	Interpreter
handle_mm_faultdo_page_faultcall_stubdo_page_faultpage_faultJavaCalls::call_helperdo_page_faultOverflowTaskQueueS.JavaCalls::call_virtualpage_faultInstanceKlass::oopJavaCalls::call_virtualpage_faultInstanceKlass::oopJavaCalls::call_virtualParCompactionManager::push_objarrayCardTableExtension:thread_entryMarkFromRootsTask::do_itOldToYoungRootsTas.JavaThread::thread_main_innerGCTaskThread::runJavaThread::runJavaThread::runjava_startstart_threadJavaThread::run				Interpreter
do_page_fault page_fault JavaCalls::call_helper do_page_fault OverflowTaskQueueS JavaCalls::call_virtual page_fault InstanceKlass::cop JavaCalls::call_virtual ParCompactionManager::push_objarray CardTableExtension: thread_entry MarkFromRootsTask::do_it OldToYoungRootsTas JavaThread::thread_main_inner GCTaskThread::run JavaThread::run JavaThread::run				Interpreter
do_page_fault OverflowTaskQueueS JavaCalls::call_virtual page_fault InstanceKlass::oop JavaCalls::call_virtual ParCompactionManager::push_objarray CardTableExtension: thread_entry MarkFromRootsTask::do_it OldToYoungRootsTas JavaThread::thread_main_inner GCTaskThread::run JavaThread::run JavaThread::run java_start start_thread Start_thread				
page_fault InstanceKlass::oop JavaCalls::call_virtual ParCompactionManager::push_objarray CardTableExtension: thread_entry MarkFromRootsTask::do_it OldToYoungRootsTas JavaThread::thread_main_inner GCTaskThread::run JavaThread::run JavaThread::run start_thread Start_thread JavaThread::run				
ParCompactionManager::push_objarray CardTableExtension:: thread_entry MarkFromRootsTask::do_it OldToYoungRootsTas JavaThread::thread_main_inner GCTaskThread::run JavaThread::run java_start start_thread	do_page_fault			JavaCalls::call_virtual
MarkFromRootsTask::do_it OldToYoungRootsTas JavaThread::thread_main_inner GCTaskThread::run JavaThread::run java_start start_thread				
GCTaskThread::run JavaThread::run JavaThread::run start_thread				
java_start start_thread			OldToYoungRootsTas	
start_thread	GCTaskThread::run			JavaThread::run
	java_start			
	start_thread			
java	java			
all	all			

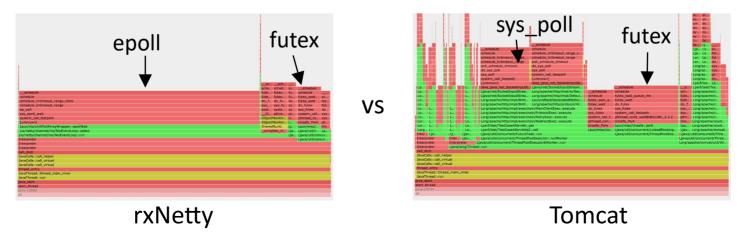
Context Switches

Show why Java blocked and stopped running on-CPU:

```
# perf record -e context-switches -p PID -g -- sleep 5
```

- Identifies locks, I/O, sleeps
- If code path shouldn't block and looks random, it's an involuntary context switch. I often filter these, but I've usually already solved this type of issue (CPU load) long before trying advanced flame graphs.

E.g., analyzing framework differences:



TCP Events

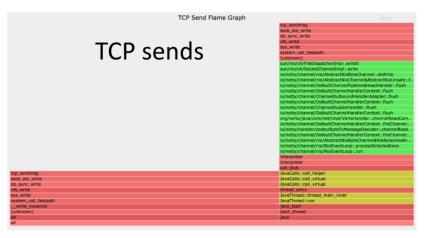
TCP transmit, using eBPF:

bpftrace -e 'kprobe:tcp_sendmsg { @[kstack, ustack] = count(); }'

- For eBPF, can cost **noticeable overhead** for high packet rates (test and measure)
- For perf, can have *prohibitive overhead* due to the trace, dump, post-process cycle
- Note that **TCP receive is async**, so stack traces are meaningless. Trace socket read instead.

Can also trace TCP connect, accept

Lower frequency, therefore lower overhead



CPU Cache Misses

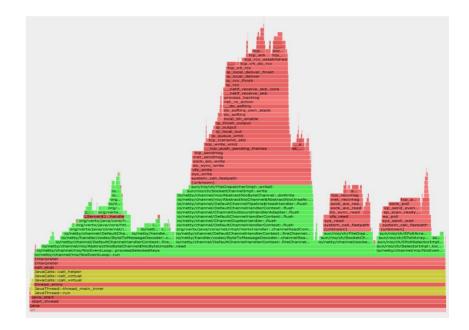
In this example, sampling via Last Level Cache loads:

perf record -e LLC-loads -c 10000 -a -g -- sleep 5; jmaps
perf script -f comm,pid,tid,cpu,time,event,ip,sym,dso > out.stacks

- -c is the count (samples once per count)
- Can also sample hits, misses, stalls

Needs PEBS for IP accuracy

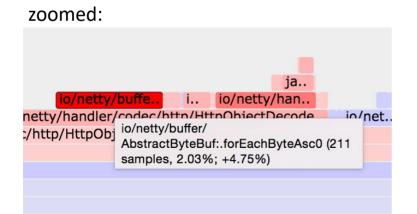
 Precise Event Based Sampling for Instruction Pointer accuracy. Not yet enabled in AWS EC2 VMs.

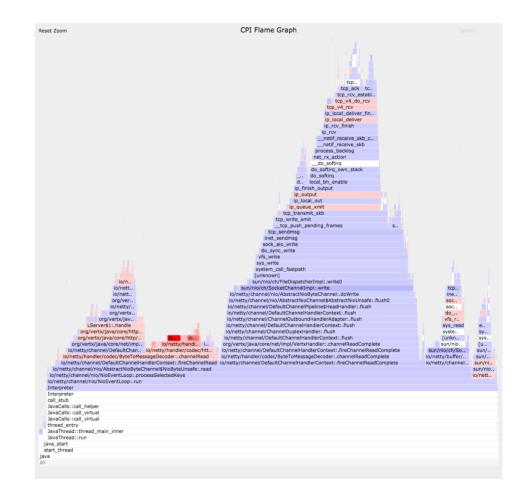


CPI Flame Graph

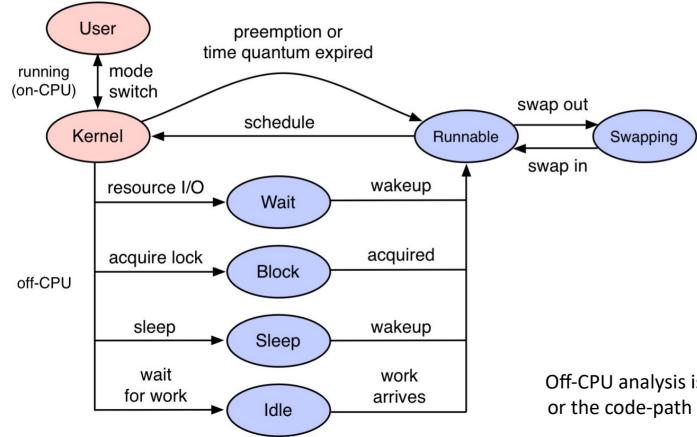
Cycles Per Instruction (CPI)

- red == instruction heavy
- blue == cycle heavy (likely memory stall cycles)



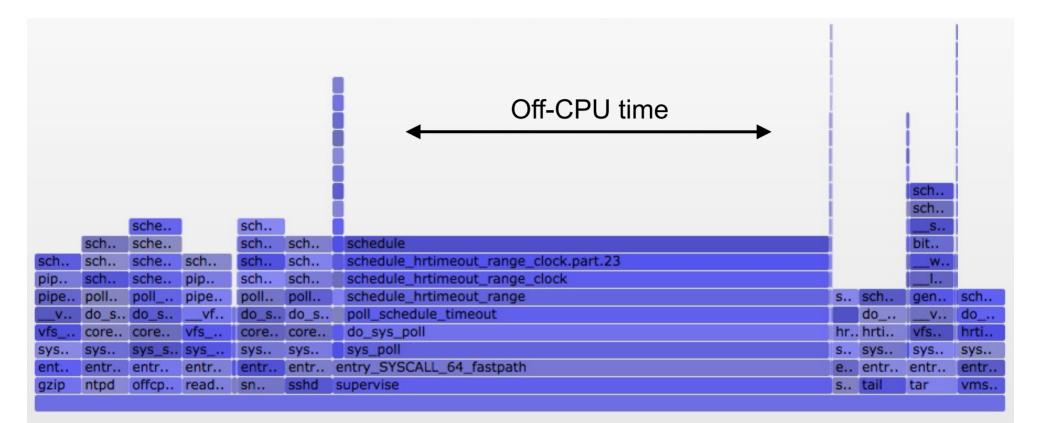


Off-CPU Analysis



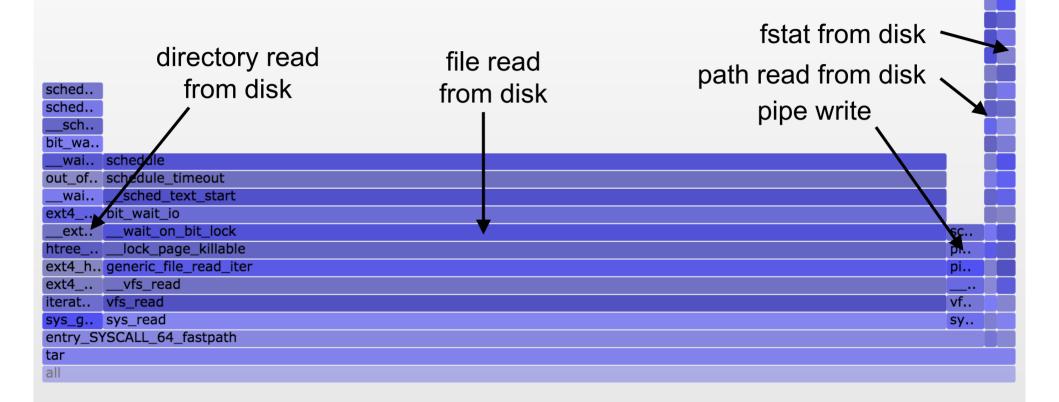
Off-CPU analysis is the study of blocking states, or the code-path (stack trace) that led to them

Off-CPU Time Flame Graph

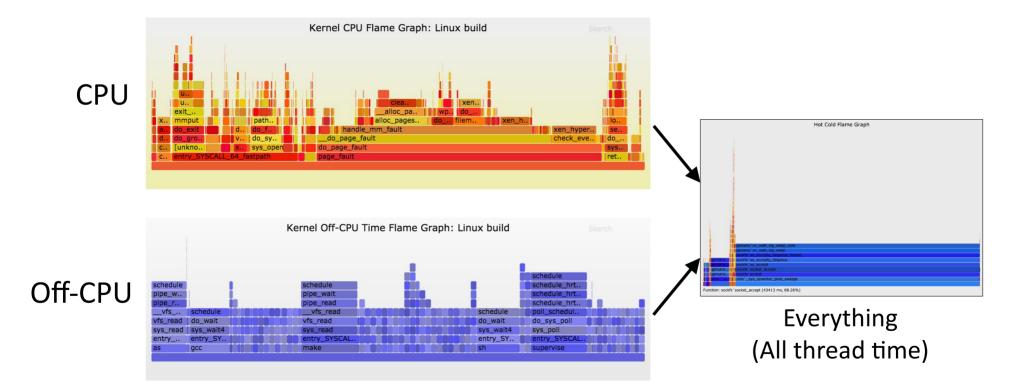


More info http://www.brendangregg.com/blog/2016-02-01/linux-wakeup-offwake-profiling.html

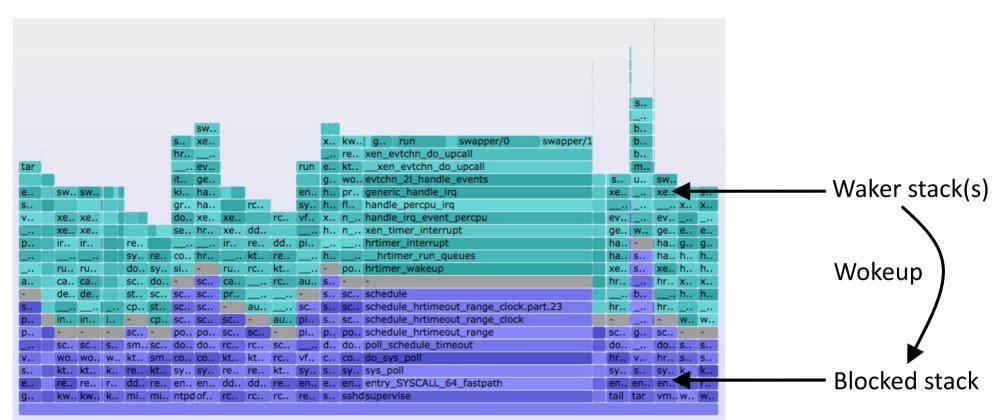
Off-CPU Time (zoomed): tar(1)



CPU + Off-CPU Flame Graphs: See Everything



Off-Wake Time Flame Graph



Uses Linux enhanced BPF to merge off-CPU and waker stack in kernel context

Chain Graphs

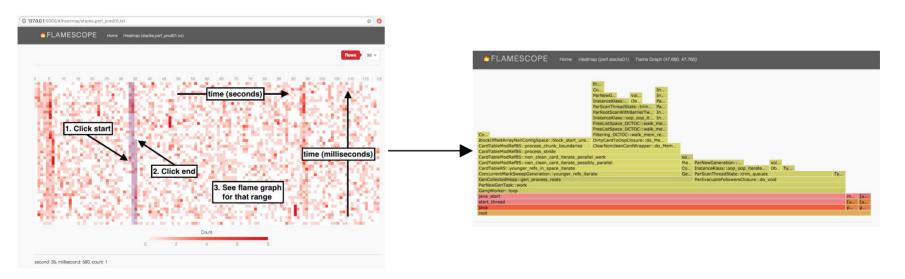


Walking the chain of wakeup stacks to reach root cause

FlameScope

Flame graphs can hide time-based issues of variation and perturbations.

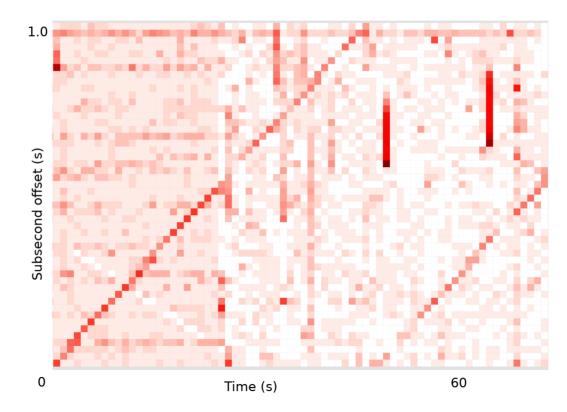
FlameScope uses **subsecond-offeset heat maps** to show these issues. They can then be selected for the corresponding flame graph.



https://brendangregg.com/blog/2018-12-15/flamescope-origin.html https://www.brendangregg.com/HeatMaps/subsecondoffset.html

FlameScope Example

How many patterns can you see?

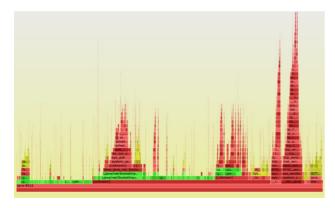


https://www.brendangregg.com/blog/2018-11-08/flamescope-pattern-recognition.html

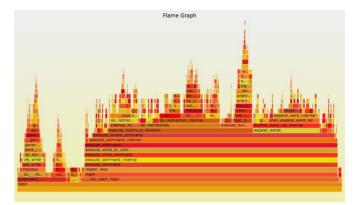
Agenda Recap



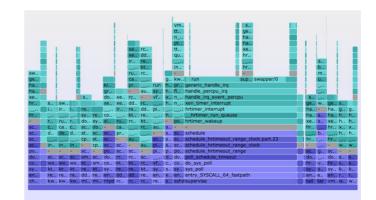
1. Implementations



3. Stacks & Symbols



2. CPU Flame graphs



4. Advanced flame graphs

Take Aways

- 1. Interpret CPU flame graphs
- 2. Understand runtime challenges
- 3. Why eBPF for advanced flame graphs

A new tool to lower your cost, latency, and carbon

Links & References

Flame Graphs

- "The Flame Graph" Communications of the ACM, Vol. 56, No. 6 (June 2016)
- <u>http://queue.acm.org/detail.cfm?id=2927301</u>
- http://www.brendangregg.com/flamegraphs.html
- http://www.brendangregg.com/flamegraphs.html#Updates
- <u>http://www.brendangregg.com/FlameGraphs/cpuflamegraphs.html</u>
- <u>http://www.brendangregg.com/FlameGraphs/memoryflamegraphs.html</u>
- <u>http://www.brendangregg.com/FlameGraphs/offcpuflamegraphs.html</u>
- http://techblog.netflix.com/2015/07/java-in-flames.html
- http://techblog.netflix.com/2016/04/saving-13-million-computational-minutes.html
- http://www.brendangregg.com/blog/2014-11-09/differential-flame-graphs.html
- <u>http://www.brendangregg.com/blog/2016-01-20/ebpf-offcpu-flame-graph.html</u>
- http://www.brendangregg.com/blog/2016-02-01/linux-wakeup-offwake-profiling.html
- http://www.brendangregg.com/blog/2016-02-05/ebpf-chaingraph-prototype.html
- https://brendangregg.com/blog/2018-12-15/flamescope-origin.html
- https://github.com/brendangregg/FlameGraph
- https://github.com/spiermar/d3-flame-graph
- https://github.com/Netflix/flamescope
- http://corpaul.github.io/flamegraphdiff/
- https://www.intel.com/content/www/us/en/develop/documentation/vtune-help/top/reference/user-interface-reference/window-fla me-graph.html
- https://gprofiler.io/
- Linux perf
 - <u>https://perf.wiki.kernel.org/index.php/Main_Page</u>
 - <u>http://www.brendangregg.com/perf.html</u>

Linux eBPF

https://ebpf.io/ https://www.brendangregg.com/ebpf.html

These slides: https://www.brendangregg.com/Slides/YOW2022_flame_graphs.pdf

YOW! 2022

nf_..

ipv4..

nf_hook_.. ip_finish ip loca.. ip outpu

nf iter..

ip local out

tcp_push_pending_frames

cp write xmit

tcp send fin

tcp close

inet release

a., sock release

inet6 release

ip_queue_xmit tcp_transmit_skb

do softirg.part.18

ip finish output2

local bh enable ip

systems

Second Edition

Performance

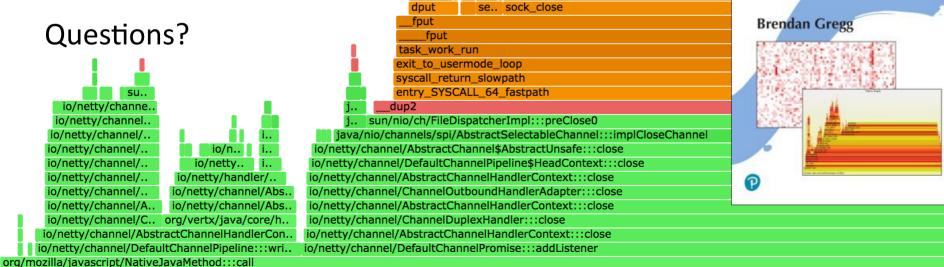
Enterprise and the Cloud

Thank you!

http://www.brendangregg.com

brendan@intel.com

@brendangregg



ev..

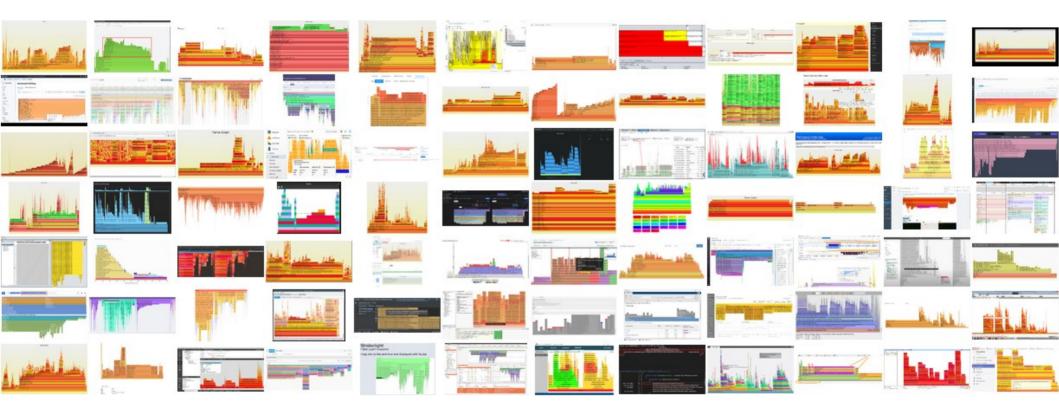
den.

iput

dentr.

BONUS SLIDES

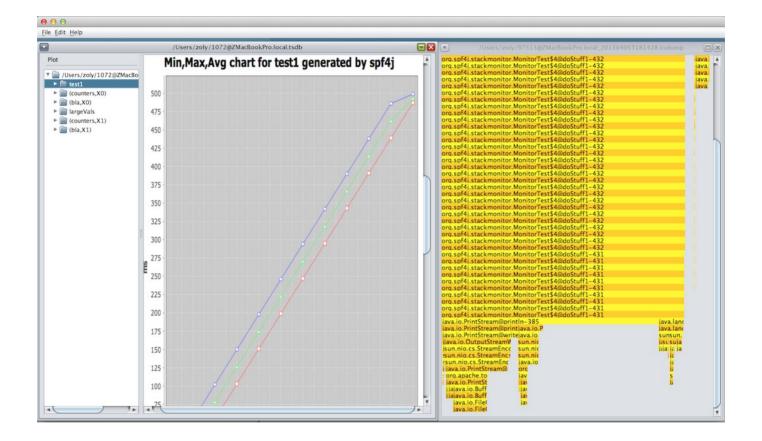
More Implementations



These are in addition to the earlier examples.

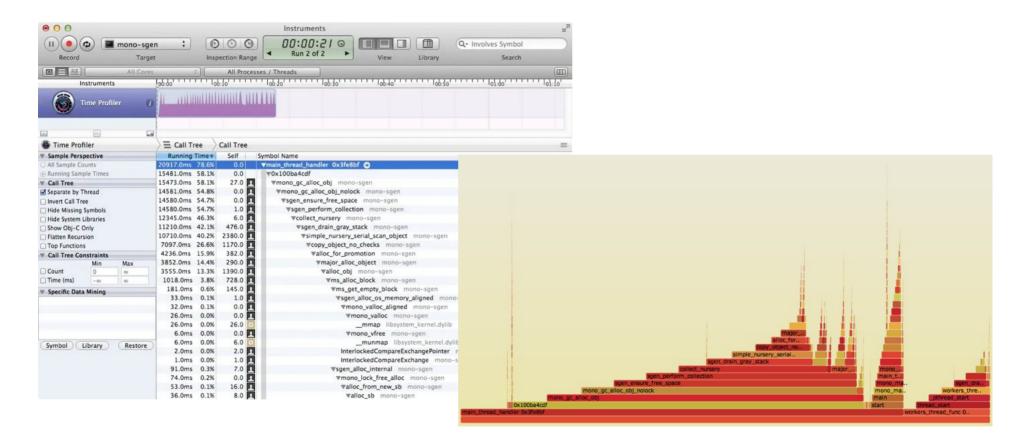
(Note: This is not an an endorsement of any company/product or sponsored by anyone.)

Java: SPF4J (2012)



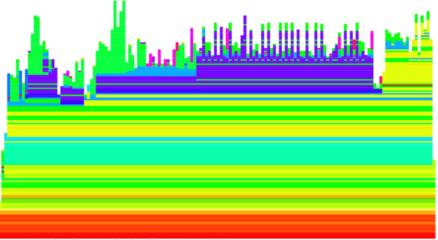
Source: http://zolyfarkas.github.io/spf4j/#

OSX: Instruments (2012; converter)



Source: https://schani.wordpress.com/2012/11/16/flame-graphs-for-instruments/

Ruby: mini-profiler (2013)

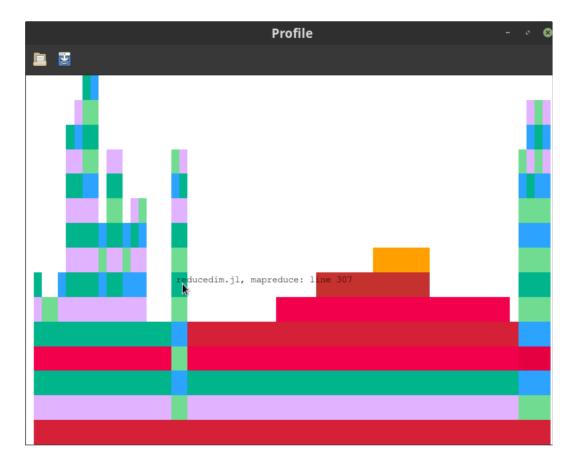


home/sam/Source/MiniProfiler/Ruby/lib/mini_profiler/profiler.tb:277:in `calf' (148 samples - 99.33%)

ruby-2.0.0-p0-turbo (148 samples - 99.33%)	thin-1.5.0 (148 samples - 99.33%)	eventmachine-1.0.1 (148 samples - 99.33%)	railties-3.2.12 (148 samples - 99.33%)	Ruby (148 samples - 99.33%)	actionpack-3.2.12 (148 samples - 99.33%)
rack-1.4.5 (147 samples - 98.66%)	silence_logger.rb:19.in 'calf' (147 samples - 98.66%)	quiet_logger.tb:10in 'call_with_quiet_assets' (147 samples - 98.66%).	activesupport-3.2.12 (147 samples - 98.663s)	activerecord-3.2.12 (147 samples - 98.66%)	message_bus (145 samples - 97.32%)
omriauth-1.1.1 (145 samples - 97.32%)	omniauth-browserid- af62d667626c (145 samples - 97.32%)	journey-1.0.4 (145 samples - 97.32%)	discourse (144 samples - 96.64%)	redis-3.0.2 (3 samples - 2.01%)	redis-activesupport- 3.2.3 (1 sample - 0.6795)
redis-store+1.1.3 (1 sample - 0.67%)	multi_json=1.6.1 (116 samples = 77.85%)	active_model_serializers= 0114e492388f(112 samples = 75.17%)	activemodel-3.2.12 (9 samples - 6.04%)	arel-3.0.2 (4 samples - 2.68%)	i18n-0.6.1 (9 samples - 6.04%)
mail-2.4.4 (5 samples - 3.36%)	tzinfo-0.3.37 (3 samples - 2.01%)	redcarpet-2.2.2 (1 sample - 0.67%)			

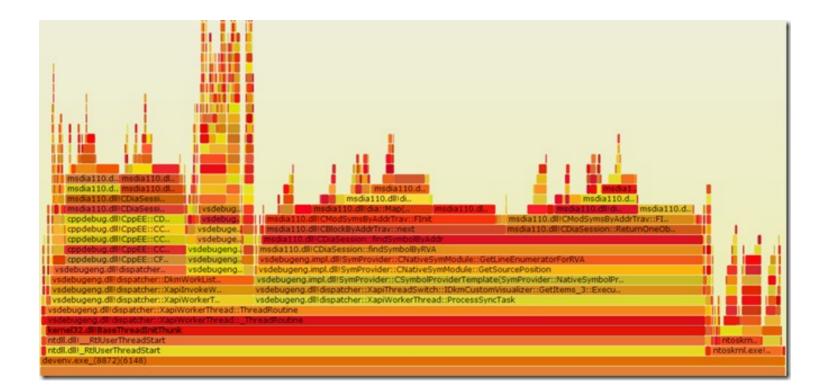
Source: https://samsaffron.com/archive/2013/03/19/flame-graphs-in-ruby-miniprofiler

Julia: ProfileView.jl (2013)



Source: https://github.com/timholy/ProfileView.jl (Tim Holy)

Windows: Xperf (2013; converter)



Source: https://randomascii.wordpress.com/2013/03/26/summarizing-xperf-cpu-usage-with-flame-graphs/ (Bruce Dawson)

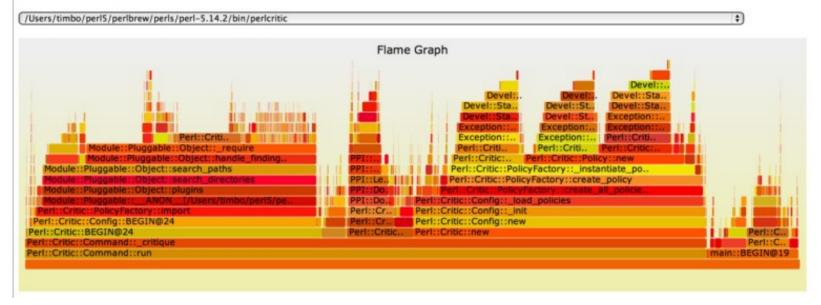
Perl: NYTProf (2013)

Performance Profile Index

For /Users/timbo/perl5/perlbrew/perls/perl-5.14.2/bin/perlcritic

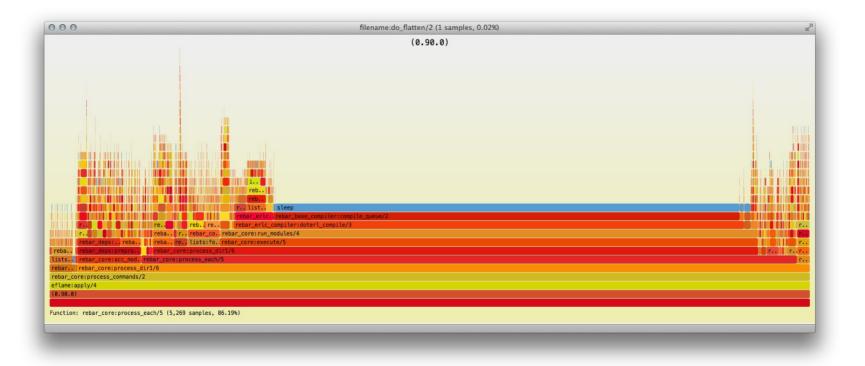
Run on Sat Apr 6 15:30:17 2013 Reported on Sat Apr 6 15:32:30 2013

Profile of /Users/timbo/perl5/perlbrew/perls/perl-5.14.2/bin/perlcritic for 1.11s (of 1.26s), executing 455869 statements and 141979 subroutine calls in 421 source files and 195 string evals.



Source: https://blog.timbunce.org/2013/04/08/nytprof-v5-flaming-precision/ (Tim Bunce)

Erlang: Eflame (2013)



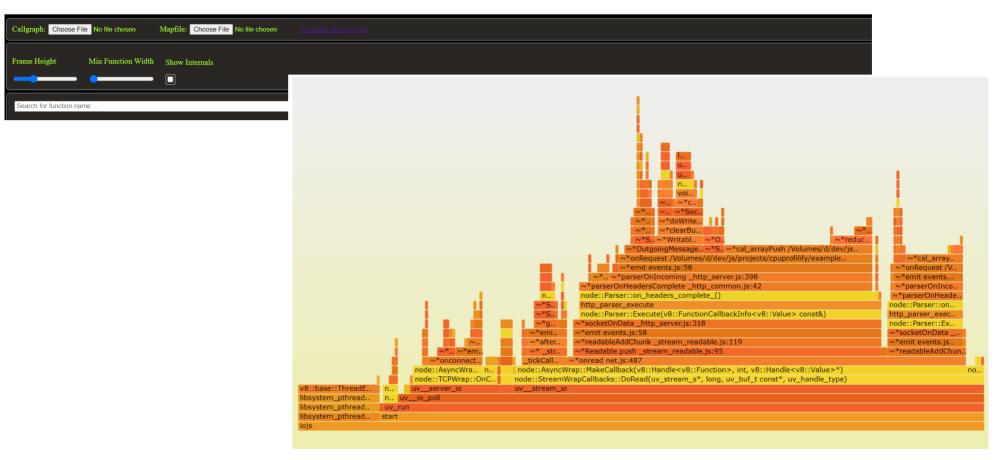
Source: https://github.com/proger/eflame (Volodymyr Ky)

Ruby: ruby-prof-flamegraph (2014)

Flame Graph	
Integer#upto (200)	
Object#is_prime (200)	
Array#select (1)	A
Object#find_primes (1)	0
Object#run_primes (1)	
Global#[No method] (1)	
Fiber:70288436678500	
Thread:70288440441800	

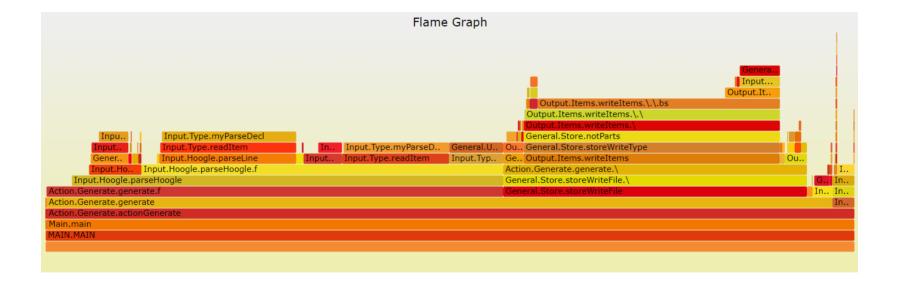
Source: https://github.com/oozou/ruby-prof-flamegraph

Node.js: flamegraph (2015)



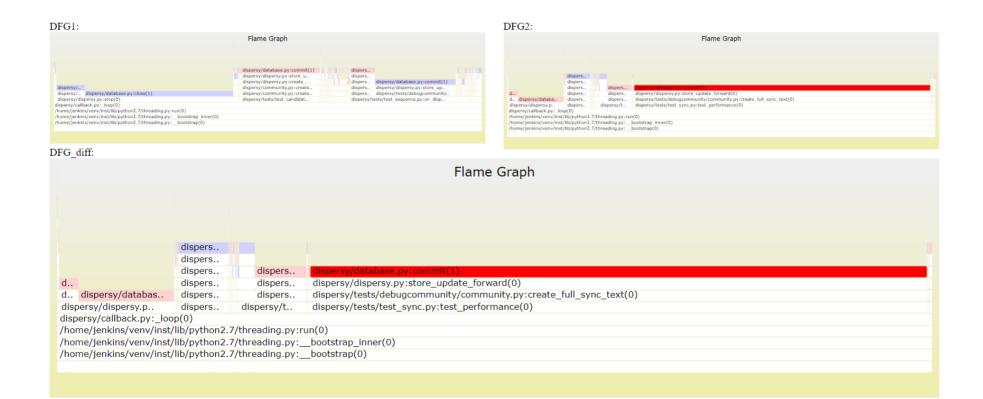
Source: https://github.com/thlorenz/flamegraph (Thorsten Lorenz)

Haskell: ghc-prof-flamegraph (2015)



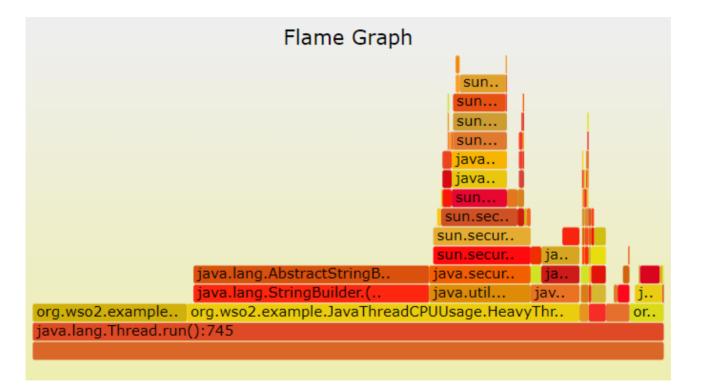
Source: https://www.fpcomplete.com/blog/2015/04/ghc-prof-flamegraph/ (Francesco Mazzoli)

Differentials: Flamegraphdiff (2015)



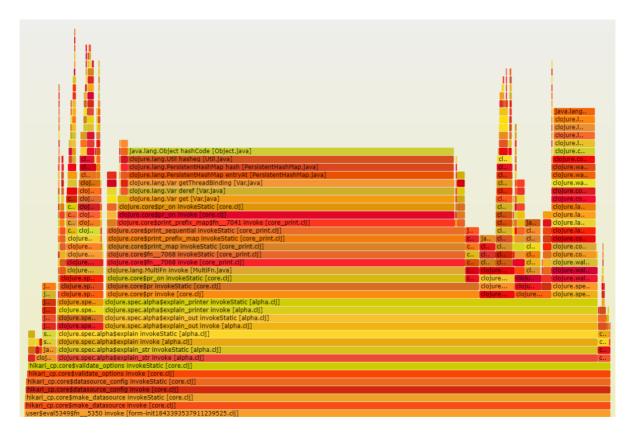
Source: http://corpaul.github.io/flamegraphdiff/ (Cor-Paul Bezemer)

Java: jfr-flame-graph (2015)



Source: http://isuru-perera.blogspot.com/2015/05/flame-graphs-with-java-flight-recordings.html (M. Isuru Tharanga Chrishantha Perera)

Clojure: Flames (2015)



Source: https://github.com/jstepien/flames/ (Jan Stępień)

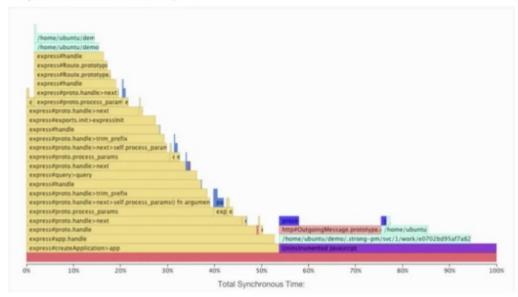
Python: python-flamegraph (2015)

	yCanta PDF			CP Server T
CP. CP Server Thread-5 print line	CP. CP Serv. CP Server T. CP. CP Server T. CP. CP Server Thread CP Server Thread-5' setFont	CP Server Thread-5' getInternal	CP Server T	CP Server T. CP Server T. CP Server T. CP Server T. CP Server T.
CP. CP Serv. CP Server Thread-5`format_page			CP Server Thread-5`parse	CP Server T.
CP Server Thread-5`format				
CP Server Thread-5`format2pdf				
CP Server Thread-5`export2pdf				
CP Server Thread-5`require				
CP Server Thread-5`export2pdf				
CP Server Thread-5`try_call				
CP Server Thread-5`_execute_func				
CP Server Thread-5`(lambda)				
CP Server Thread-5`_expose				
CP Server Thread-5`_expose				
CP Server Thread-5`so_rwt CP Server Thread-5`run with transaction				
CP Server Thread-5 run_with_transaction CP Server Thread-5`expose				
CP Server Thread-5 expose CP Server Thread-5 export2pdf				
CP Server Thread-5° call				
CP Server Thread-5can CP Server Thread-5 respond				
CP Server Thread-5`run				
CP Server Thread-5`run				
CP Server Thread-5` init				
CP Server Thread-5 'tail				
CP Server Thread-5` call				
CP Server Thread-5' trap				
CP Server Thread-5 init				
CP Server Thread-5`call				
CP Server Thread-5`call				
CP Server Thread-5`call				
CP Server Thread-5°call				
CP Server Thread-5`respond				
CP Server Thread-5`respond				
CP Server Thread-5° communicate				
CP Server Thread-5`run				
CP Server Thread-5` bootstrap inner				

Source: https://github.com/evanhempel/python-flamegraph (Evan Hempel)

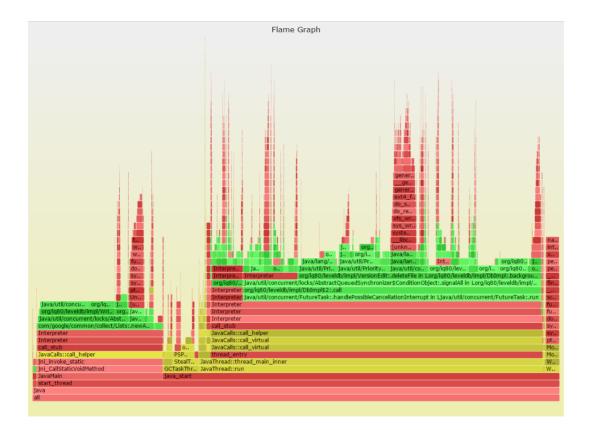
Strongloop: Arc (2015)

Synchronous Code Flame Graph @



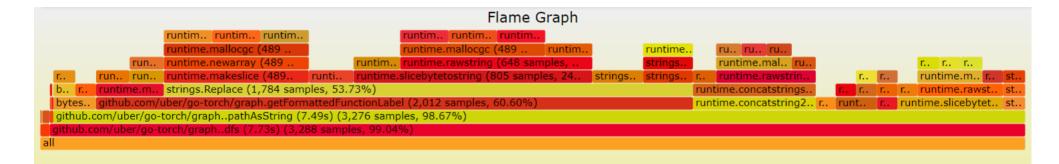
Source: https://es.slideshare.net/jguerrero999/nodejs-transaction-tracing-root-cause-analysis-with-strongloop-arc

Java: perfj (2015)



Source: https://github.com/coderplay/perfj (Min Zhou)

Golang: Uber go-torch (2015)



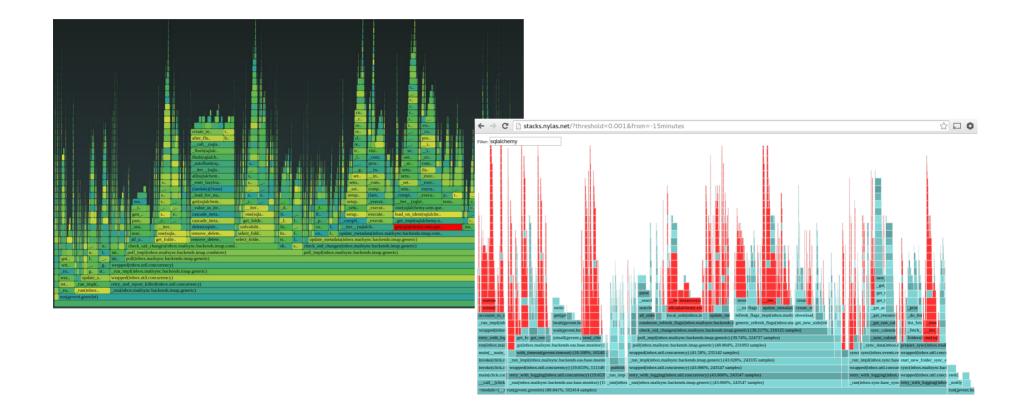
Source: https://github.com/uber-archive/go-torch

Intel: processor trace converter (2015)

	ite.///ite	me/	ак/рпп	u/x.svg								5		9	-
									Flame Graph						
			ICAE DIVINIO	st.			81100				Iunk.	[unknown]			
T.											THE OWNERS CONTRACTOR OF THE		11-1-11		
				do aut			TTT		do autocmd			do cmdline	fu.		
L.				do_cmdline		w	100		do cmdline		10_f.	GIlibc_read	[un	111	
				_IO_fgets					GIlibc_read		[unk	_IO_file_underflow@@GLIB	[un	111	
d				[unknown]		NT			IO_file_underflow@@GLI		getsou	_IO_default_uflow	vim	11	
				getsource					_IO_default_uflow		do_cm	_10_getline_info	[un		
1				do_cmdline		NT.			_IO_getline_info		do_sou	_IO_fgets	do		
ge				do_source					_IO_fgets		ga_grow	[unknown]	do		
do.				ga_grow		and a			[unknown]		do_sou		_IO		
do				do_source							do_sou		[un		
						na.									g.
	g.,					-			2 March 1997 Control of Control o		Contraction of the local division of the loc			_	
do			-								and the second se		ga_g		
			_1.	xstat64		d		do	cmdline		do_source				
	[]	Image: Section of the section of t			L. (Intervention of the second	Image:	Image:	Image: Strain	Image: State Stat	Image: Structure Structur	Image: Structure Structur	Image: Structure Structur	Image: Structure Structur	Image: Figure Set Strictup: Set Stricup: Set Strictup: Set Strictup: S	Image: Structure Structur

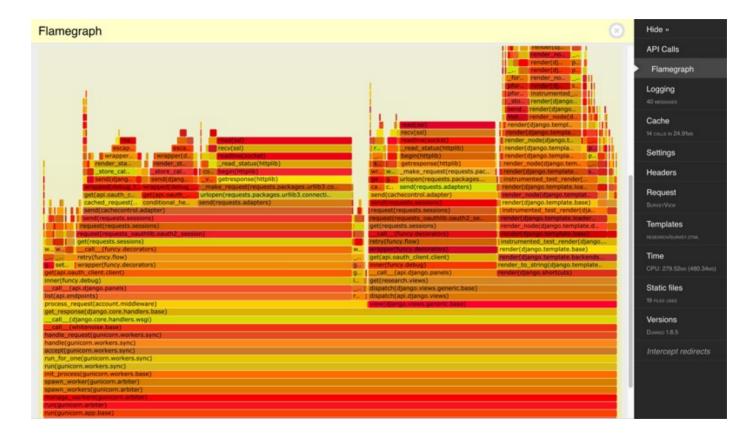
Source: http://halobates.de/blog/p/329 (Andi Kleen)

Nylas: perftools (2015)



Source: https://www.nylas.com/blog/performance/ (code by Eben Freeman)

Django: djdt-flamegraph (2015)



Source: https://github.com/blopker/djdt-flamegraph (Bo Lopker)

NodeSource: Nsolid (Node.js; 2015)



Source: https://nodesource.com/blog/understanding-cpu-flame-graphs

D3: d3-flame-graphs (2015)

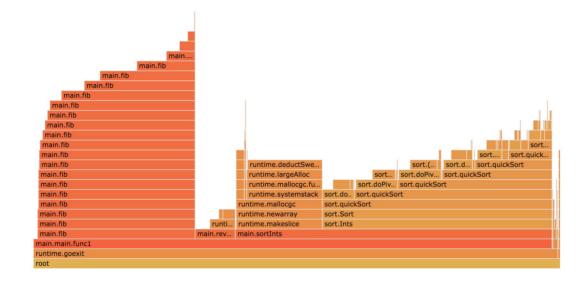
D3 flame graph plugin Actions -

🖸 GitHub

		Unsafe.park LockSupport.parkNanos			
		SynchronousQueue\$TransferStack.awaitFulfill			
		SynchronousQueue\$TransferStack.transfer			
		SynchronousQueue.poll			
	ThreadPoolExecutor.getTask			17	
	ThreadPoolExecutor.runWorker			1DĪ	
	ThreadPoolExecutor\$Worker.run				
Т	hread.run				
ALL.					

Source: https://cimi.io/d3-flame-graphs/ (Alex Ciminian)

Golang: Goprofui (2015)



Source: https://github.com/wirelessregistry/goprofui (Srdjan Marinovic, Julia Allyce)

Rust: flame (2016)

filter poin		filter po		
collect sam	gather lines	collect sa	gather lines	
render		render		
entire thing				

Source: https://github.com/llogiq/flame (Ty Overby)

Dell Cloud Manager: Gumshoe Load Investigator (2016)

¢		(on delitower)					
Arr Bits Tax Paris Bits Tax Paris Stat Tax Paris Stat Tax Paris Stat Tax Bar Stat Tax Bar Perry Tax Bar Tax Data Default Tax Data Default Tax System Service Default Tax System Service agetSt Default Tax System Service agetSt Default Tax System Service agetSt Default Tax System Service agetSt Default Default Tax System	CurCu Acti En Gra CurCu CurSan En Gra CurCu CurSan En Gra CurCu CurSan CurCu ActionEmstrated CurAcistack of Enstmemstrated CurAcistack of Enstmemstrated		Image: Start Factor Start Start Factor Start	2			Def
ApiOnginFilter.doFilter.19			ExecutingTeskServer.ExecutingTesk.run:1266 BaseAuthoFiltenE	xedrog Di	Database.getDocument:213	Statement.execute:118	StolTransaction.commit:2477
Collect -> -> Filter -> -> Display -> -> Ex	amine				CatalogDAO.getitemimages:118	CatalogDAO.getitemDetails:118	InvePaymentManager.confirmL
() read	⊖ write		(e) read+write				
Measurement					ShoppingRequestHandler.showitemDetails:431		ShoppingRequestHandleL
ops) bytes		 time(ms) 				
Direction Show called methods (flame graph)		🔾 show ci	allers (root graph)		RestRequestHandler.handle:131		
Cell width-					Invoker.invoke:281		
"This haystack is lookir	ng more like a needle ev	ery minute"	source: https://youtu.be/GGJFZfwXJ44?t=	225			
,	5	,	, ,,, , , , , , , , , , , , , , , , , ,	-		1	

Collect --> --> Filter --> Display --> --> Examine

💿 read

ops/count

write

bytes

Select statistic to display For sample type [socket-io]

Operation:

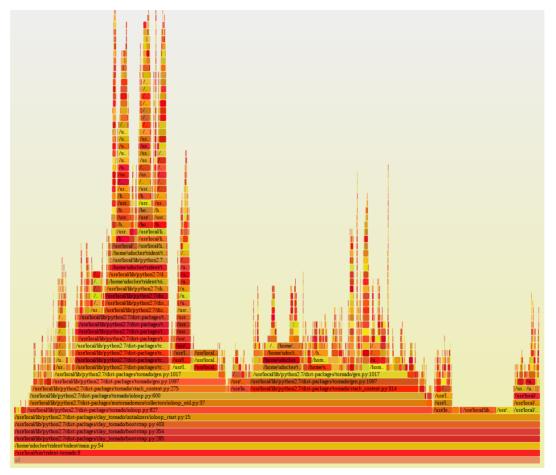
Measurement

Source: https://github.com/worstcase/gumshoe (Jonathan Newbrough)

read+write

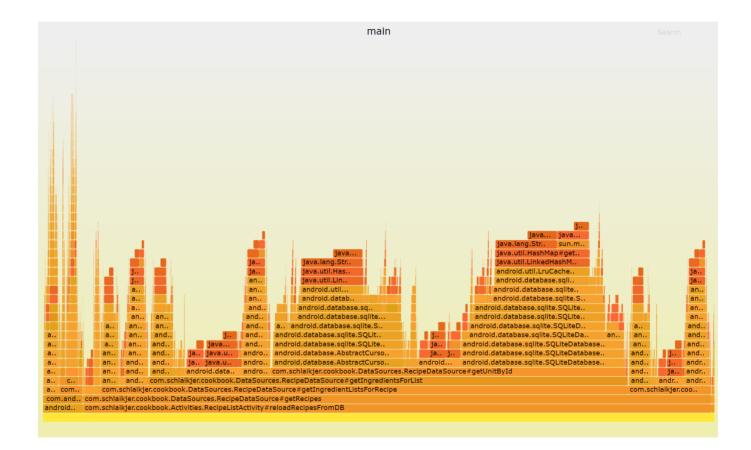
O time(ms)

Uber: pyflame (Python; 2016)



Source: https://www.uber.com/en-AU/blog/pyflame-python-profiler/

Android: erlang-atrace-flamegraphs (2017)



Source: https://blog.rhye.org/post/android-profiling-flamegraphs/ (Ross Schlaikjer)

Java: grav (heap allocations; 2017)

	org/eclips	org/e	ог	0			org/e	0	OF	0	org/e
	org/eclips	org/e	ог	0			org/e	0	OF	0	org/e
	org/eclips	org/e	ог	0			org/e	0	OF	0	org/e
	org/eclips	org/e	ог	0			org/e	0	OF	0	org/e
	com/codaha	org/e	ог	0			org/e	0	OF	0	org/e
	io/dropwiz	com/c	co	C.,			com/c	0	CO	C	com/c
	org/eclips	io/dr	io	i.,			io/dr	с	io	i.,	io/dr
	io/dropwiz	org/e	ог	0			org/e	i	OL.	0	org/e
	org/eclips	io/dr	io	i.,			io/dr	0	io	i.,	io/dr
	org/eclips	org/e	ог	0			org/e	i.,	OF	0	org/e
	org/eclips	org/e	ог	0			org/e	0	OF	0	org/e
	org/eclips	org/e	ог	0			org/e	0	OF	0	org/e
	org/eclips	org/e	ог	0			org/e	0	OF.	0	org/e
	org/eclipse	org/e	ог	0		0	org/ec	0	org	0	org/ec
	org/eclipse	org/e	OT	0		0	org/ec	ог	org	0	org/ec
	org/eclipse	org/e	or	0		0	org/ec	ог	org	0	org/ec
	org/eclipse	org/e	OГ	0		0	org/ec	ог	org	0	org/ec
	org/eclipse/	org/e	or	0		0	org/ec	ΟГ	org	0	org/ec
	org/eclipse/	org/e	org	0	0	ог	org/ecl	ог	org/	0	org/ec
	org/eclipse/	org/e	org	0	0	ΟГ	org/ecl	ог	org/	0	org/ec
	org/eclipse/	org/e	org	0	0	ог	org/ecl	ог	org/	0	org/ec
	org/eclipse/jetty/u		org	0	0	ОГ	org/ecl	ог	org/	0	org/ec
. 0	Interpreter		org/e	org	org	org/e	org/eclipse		org/ec	org	org/eclips
nter	call_stub		Inter	Int	Int	Inter.	Interpreter		Interpr	Int	Interpreter
all	JavaCalls::call_he	per	call	cal	cal	call	call_stub		call_stub	cal	call_stub
avaC	JavaCalls::call_vir	t.,	JavaC	Jav	Jav	JavaC	JavaCalls::c		JavaCal	Jav	JavaCalls:
avaC	JavaCalls::call_vir	t	JavaC	Jav	Jav.,	JavaC	JavaCalls::c		JavaCal	Jav.,	JavaCalls:
avaC	thread_entry		JavaC	Jav	Jav	JavaC	JavaCalls::c		JavaCal	Jav	JavaCalls:
nrea	JavaThread::thread	l_m	threa	thr.	thr	threa	thread_entry		thread	thr	thread_ent
avaT	JavaThread::run		JavaT	Jav	Jav	JavaT	JavaThread::.		JavaThr	Jav	JavaThrea
avaT	java_start		JavaT	Jav	Jav	JavaT	JavaThread::.		JavaThr	Jav	JavaThrea
ava	start_thread		java	jav	jav	java	java_start		java_st	jav	java_start
tart			start	sta	sta	start	start_thread		start_t	sta	start_threa
ava/	java/lang/Object[]		java/	jav	jav	java/	java/util/Ha		java/ut	jav	java/util/

Source: https://epickrram.blogspot.com/2017/09/heap-allocation-flamegraphs.html (Mark Price)

Nudge: APM (for Java; 2017)

Overview O Map Transactions Services	A User Sessions	
Overview Profiling NEW Errors (Details		
SELECTED ITEM Class: Local time: 0.00 % Total time: 60.07 %	Line: 200 TRANSACTION Executions: 609428 Mean Time: 855.71 ms Samples: 387347 Nudge Overhead: 0.00 %	
CALL GRAPH	METHODS BY TIME SPENT LOCALLY Class ↓↑ Method ↓↑ Line	Local tir
	sun.misc.Unsafe park -2	79.81%
	java.lang.Class forName0 -2	4.58%
	java.lang.System identityHashCode -2	1.78%
	sun.nio.ch.Native current -2	1.23%
	java.util.ArrayList indexOf 317	1.19%
	sun.nio.ch.EPollA interrupt -2	1.01%
	org.HdrHistogram autoAdjustRange 393	0.76%
	un.misc.Unsafe park1-2 sun.nio.ch.FileDis pread0 -2	0.71%
sun sun.misc.Unsafe parkı-2 jav java.util.concurrent	ava.util.concurrent.lock ava.util.concurrent.lock sun.misc.Unsafe unpark -2	0.48%
jav java.util.conchrent su jav java.util.conchrent ja jav java.util.conchrent	ava.util.concurrent.lock ava.util.concurrent.lock java.lang.Object hashCode -2	0.44%
ja com com.google.common.ut 0 ja com com.google.common.ut 0	om.google.common.util.co java.lang.Object notifyAll -2	0.26%
co com. com.google.common.ut 0 co com.google.common.ut c com.com.com.com.com.com.com.com.com.com.	om google versee util.co an udge agen datas segentus i java.util.regex.Pat match 3776	0.25%
	.collector PayDetaBrvd pr	0.24%
Analyse rawdata	iava util zin Deflater deflateBytes -2	J.L. 170

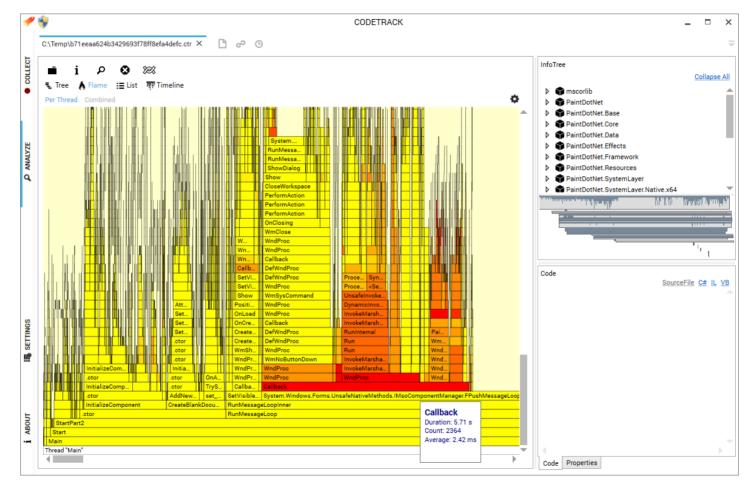
Source: https://nudge-apm.com/features/#profiling

Java: clj-async-profiler (2017)

	Flame Graph	
	java/math/MutableBigInteger.divideOneWord	
Let a set of the set o	java/math/MutableBigInteger.divideOneword	
java/math/MutableBigInteger.divideOneWord	java/math/MutableBigInteger.divide	
java/math/MutableBigInteger.divideOneword	java/math/MutableBigInteger.divide	jlong_d.
java/math/BigInteger.divideKnuth	java/math/MutableBigInteger.hybridGCD	jong_d
java/math/BigInteger.divide	j., java/math/BigInteger.gcd	
clojure/lang/Numbers.divide	j., java/math/BigInteger.gcd	java/math/BigInteger.mul
clojure/lang/Numbers.divide		Java/math/biginteger.mul
clojure/lang/Numbersskatdops.dvide		
clojure/lang/Numbers.divide clojure/core\$_SLASHinvokeStatic		
clojure/core\$_SLASH_invokeStatic		
clojure/cores_SLASH_invoke clojure/lang/ArrayChunk.reduce		
clojure/lang/ArrayChunk.reduce clojure/core/protocols\$fn 7847.invokeStatic		
clojure/core/protocols\$fn_7847.invoke clojure/core/protocols\$fn_7807\$G_7802_7816.invoke		
clojure/core/protocols\$seq_reduce.invokeStatic		
clojure/core/protocols\$fn7835.invokeStatic		
clojure/core/protocols\$fn7835.invoke		
clojure/core/protocols\$fn7781\$G77767794.invoke		
clojure/core\$reduce.invokeStatic		
clojure/core\$reduce.invoke		
user\$test_div.invokeStatic		
user\$test_div.invoke		
user\$burn_cpu.invokeStatic		
user\$burn_cpu.invoke		
user\$eval235.invokeStatic		
user\$eval235.invoke		
clojure/lang/Compiler.eval		
clojure/lang/Compiler.eval		
clojure/core\$eval.invokeStatic		
clojure/core\$eval.invoke		
:lojure/main\$repl\$read_eval_print8572\$fn8575.invoke		
:lojure/main\$repl\$read_eval_print8572.invoke		
:lojure/main\$repl\$fn8581.invoke		
:lojure/main\$repl.invokeStatic		
:lojure/main\$repl_opt.invokeStatic		
:lojure/main\$main.invokeStatic		
:lojure/main\$main.doInvoke		
:lojure/lang/RestFn.invoke		
:lojure/lang/AFn.applyToHelper		
:lojure/lang/RestFn.applyTo		
lojure/lang/Var.applyTo		
iojure/iang/vanappiyro		
:lojure/main.main		

Source: http://clojure-goes-fast.com/blog/profiling-tool-async-profiler/ (Alexander Yakushev)

.NET: codetrack (2017)



Source: https://www.getcodetrack.com/

Node.js: Flamebearer (2018)

💏 flamebearer

Search... Reset view

nonymous) bootstrap (unknown) (6.79%, 2 (builtin) PromiseHandle (82.72%, 3	303 of 3993 samples)							
artup bootstrap_node	~(anonymous) ./dist/rollup.js:20116	6:2~(anonymous) ./dist/rollup.js:19762:61	(17.96%, 717 of ~(anonymous) ./dis	st, ~(ano ~(anonymo	us) ./dist/rollup.js:171	90:41 (31.71%, 12	66 of 3993 samples)	~(anonymous) ./n	~(anonymous
odule.runMain modul	~Module.setSource ./dist/rollup.js:1	165 ~Graph.includeMarked ./dist/rollup.js:19	9733:46 ~Graph. ~Chunk.preRend	~res ~transform	./node_modules/rollup	o-plugin-typescript/	dist/rollup-plugin-typescript.cjs.js:268:33 (31.4	~checkE ~transform	~gen ~collap:
_tickCallback interna	~tryParse ./dis~Module.analyse	~Module.includeInBundle ./dist/n Modul	le.includ ~Modul ~Module.rend	~noi ~transpileM	odule ./node_module	s/typescript/lib/type	escript.js:93318:29 (30.35%, 1212 of 3993	~tryPars ~wa	(built
C++) node:: (7.04%,	~parse ./dist/r ~enhance ./dist/	~VariableDecl Node V	/ariable NodeBa ~Progr ~clone	~noi ~createS	~emit ./node_modul	es/typescript/lib/typ	pescript.js:76232:22 (26.27%, 1049 of 399.	~parse .	~(an
builtin) PromiseHandl	~parse ./dist/ enhancelNodeBa	(builtin) Array (I	builtin) NodeB	~try ~parseSc	~runWithCancellation	Token ./node_mod	dules/typescript/lib/typescript.js:76322:42 (~parse .	
~(anonymous)	~pp\$1.parseT enhancel	~(anonymou	NodeB	~parseSc	~(anonymous) ./nod	e_modules/typescr	ipt/lib/typescript.js:76233:54 (26.27%, 104	~parse	
~transform ./no	pp\$1.pars enhancel	NodeBas	NodeE	~pars	~emitWorker ./node	_modules/typescrip	t/lib/typescript.js:76238:28 (26.27%, 1049	~pp\$1.p	
~transpileModu	~pp\$ enhance	NodeBas	NodeE		~getDiagnost~emitF	iles ./node_modu~	getEmitResolver ./node_modules/typescript		
~emit ./node	enhance	Funct	NodeE		~createType ~trans	formNo~forEach~	getDiagnostics ./node_modules/typescript/l		
~runWithCa	enhance	~Blo	NodeE		~initialize ~map	/node ~emitSol ~!	getDiagnosticsWorker ./node_modules/type		
~(anonymol	enhano	(buil	Nodel				forEacforEach ./node_modules/typescript/lil		
~emitWorke	enhanc		Nodel			_	check ~checkSourceFile ./node_modules/ty		
~getEr	enhanc		Nodel		bind ./nc ~reduc		check ~checkSourceFileWorker ./node_mod		
~getDi	enhan		Nodel		~bindCo ~(anor		forEa ~checkE forEach ./node_modules/typ		
~getDi	enhan		Node				checl ~checkF (unknc ~checkSourceE check		
~forEa	enhar		Node			sitSou ~on	~che ~checkC		
~check	enha					isitNo ~on	~che forEach		
~check	enha					~vis ~on	~che ~check!		
	enha					~on	~chec		
						~on	forE: ~chec		
						~on	~che ~che	1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 -	
						pipe	~che	A State of the second sec	
							forEa		
							~che		

Source: https://github.com/mapbox/flamebearer (Volodymyr Agafonkin)

Opsian: always-on flame graphs (2018)

Bottom-up 👻				nioEventLoopGroup-?-?	×
		Spring (6%) Uitits			
		String (6%) Little			
		WestProfProto\$FrameEntry			
		WestProfProto\$FrameEntry			
		WestProfProto%FrameEntry%1			
		WestProfProto%FrameEntry%1	PgStatement		
			PgStatement	LZ4JNI	
		WestProfProto\$StackSample (16%)			
		WestProfProto\$StackSample (16%)			
		WestProfProto\$StackSample\$1			
		WestProfProto\$StackSample\$1			
		CodedinputStream\$ArrayOecoder			
		WestProfProtoSAgentEnvelope	DbPool.db (12%)	FileStackWriterService	
		WestProfProto\$AgentEnvelope WestProfProto\$AgentEnvelope\$1		messaccomterservice	
		WestProfProto\$AgentErwelope\$1		Inter-addStack\$default (26%)	
		AbstractParser (16%)	c.s.w.c.services.ProtocolServeri		
		AbstractParser.parseFrom (16%)	c.s.w.c.services.ProtocolServeri	iandier.channelRead0 (32%)	
		AbstractParser.parseFrom (16%)	io.netty.channel.SimpleChannelInb	oundHandler.channelRead (32%)	
		AbstractParser.parseFrom (16%)	AbstractChannelHandlerConte		
		ProtobufDecoder.decode (17%)	AbstractChannelHandlerConte		
		ProtobufDecoder.decode (17%)	Ln.channel.AbstractChannelHandle		
			dier.codec.MessageToMessageDecoder.channelRe		
			el AbstractChannelHandlerContext.invokeChannel		
			el AbstractChannelHandlerContext.invokeChannel nel AbstractChannelHandlerContext.freChannelRi		ByteToMessageDecoder ByteToMessageDecoder
			dler.codec.ByteToMessageDecoder.freChannelRe		ByteToMessageDecoder
	FileDispatcherImpl		io.netty.handler.codec.ByteToMessageDecoc		
	SocketDispatcher (12%)		io.netty.channel.AbstractChannelHandlerContext		
	IOU0I (1296)		io.netty.channel.AbstractChannelHandlerContext		
	s.n.ch.IOUtiLread (12%)		io.netty.channel.AbstractChannelHandlerConte		
	SocketChannelimpl.read (14%)		io.netty.channel.DefaultChannelPipeline\$HeadC		
	PooledUnsafeDirectByteBuf		io.netty.channel.AbstractChannelHandlerContext		
	AbstractByteBuf (14%) NioSocketChannel (15%)		io.netty.channel.AbstractChannelHandlerContext io.netty.channel.DefaultChannelPipeline.fir		
Selectorimpi	recounterchannel (15%)	io netty, channel nis A	to.netty.channel.betauttchannelinpeine.tr	echanimetereta (aura)	
Selectorimpi			nio.NioEventLoop.processSelectedKey (84%)		
			ioEventLoop.processSelectedKeysOptimized (84%)		
NicEventLoop			nio.NioEventLoop.processSelectedKeys (84%)		
		io.netty.channel.nio.Nic			
		io.netty.util.concurrent.SingleThr			
		io.netty.util.concurrent.DefaultThreadFacto			

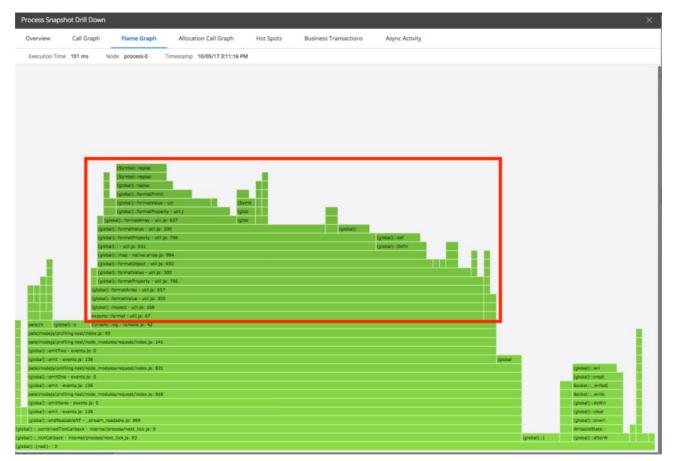
Source: https://www.opsian.com/blog/always-on-production-flame-graphs/

Speedscope: left heavy view (2018)

	🛛 Left Heavy 🏾 🎾 S			sucrase.json			🗊 Export 💽 Im	
200.00ms	400.00ms	600.00ms	800.00ms	1.00s	1	.20s	1.40s	1.6
			=			<i></i>		d ^e lener en elener el
200.00ms	400.00ms	600.00ms	800.00ms	1.00s		.20s	1.405	1.6
fulfilled	400.00ms	800.00ms	CanvasContext.or		1		rerender	(pram)
next				pView.onBeforeFrame			renderComponent	(pralli)
(anonymous)			renderRects	pv rew. onber of er fame			setComponentProps	
importProfileGroup			renderBehind				renderComponent	
awaiter			setViewport				diff	
Promise				ext.renderBehind			idiff	
(anonymous)			render				innerDiffNode	
next			writeToAtlasIfNe	eded			idiff	
(anonymous)			renderInto				innerDiffNode	
_importProfileGroup			(anonymous)				idiff	
awaiter			rowAtlas.writeTo	AtlasIfNeeded			buildComponentFromVNode	
Promise			forEachLeafNodeW	lithinBounds			createComponent	
(anonymous)			forEachLeafNodeW	lithinBounds			Connect	
next			layers.(anonymou	s funct…afNodeWithi	Bounds.leaf		initSelector	
(anonymous)			render				runComponentSelector	
importFromChromeCPU	Profile		getBuffer				pureFinalPropsSelector	
enterFrame le	aveFrame <mark>f…e</mark>						handleFirstCall	
_eme	.e						mergePropsProxy	
							mergeProps	
							exports.Chr…mechartView	
							args	
							(anonymous) expo…hart	
							Flame…derer Flamechart	
							(gr) forECall	

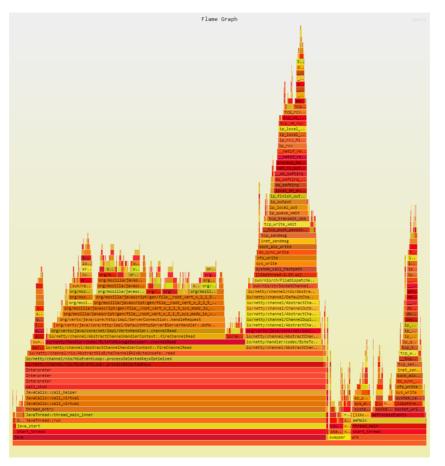
Source: https://jamie-wong.com/post/speedscope/ (Jamie Wong)

AppDynamics: flame graph (2018; now Cisco)



Source: https://docs.appdynamics.com/appd/20.x/en/application-monitoring/troubleshooting-applications/ event-loop-blocking-in-node-js#EventLoopBlockinginNode.js-FlameGraph

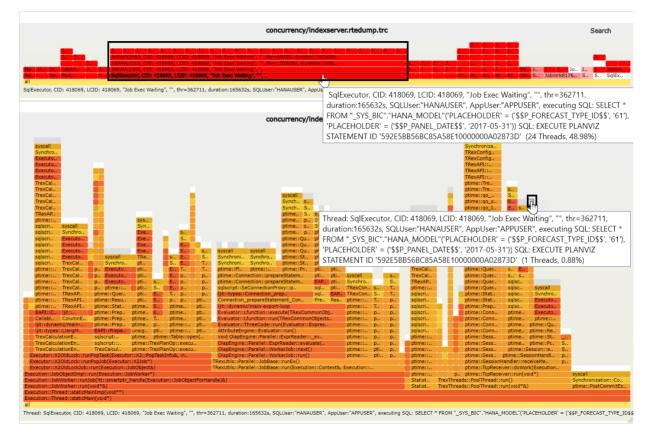
Inferno: flame graph (Rust port; 2019)



Source: https://github.com/jonhoo/inferno (Jon Gjengset)

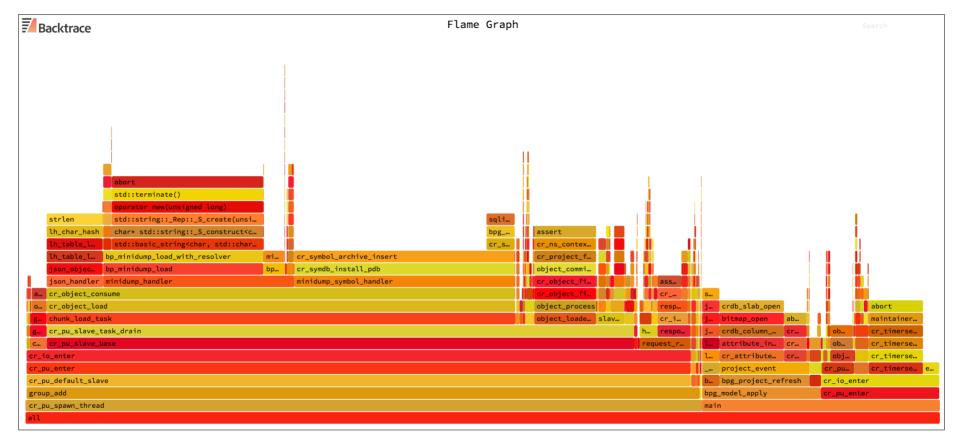
SAP: HANA Dump Analyzer (2019)

Mixed Concurrency Flame Graph



Source: https://blogs.sap.com/2019/04/22/visualizing-olap-requests-on-sap-hana-system-with-concurrencyflame-graph-using-sap-hana-dump-analyzer/

Backtrace: flame graph (2019)



Source: https://support.backtrace.io/hc/en-us/articles/360040515971-Flame-graphs

Instana: flame graph (2020; now IBM)



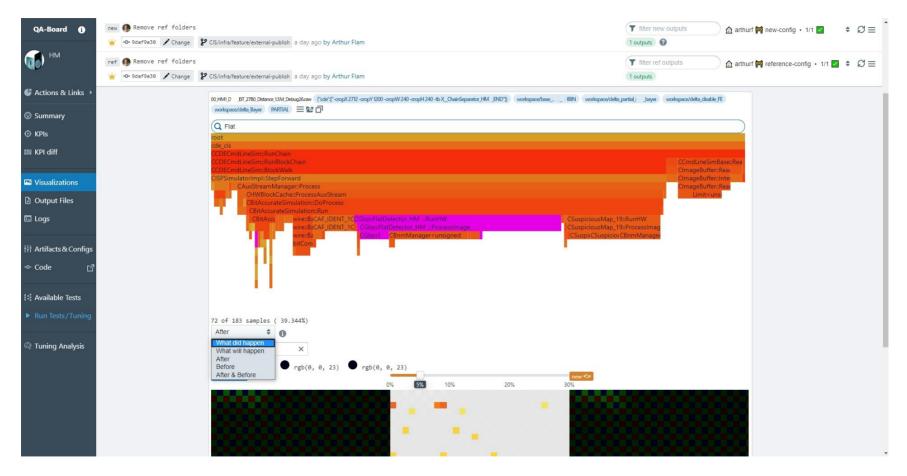
Source: https://www.ibm.com/docs/en/instana-observability/current?topic=processes-analyzing-profiles

ej-technologies: JProfiler Flame Graph (for Java; 2020)



Source: https://www.ej-technologies.com/resources/jprofiler/help/doc/main/cpu.html

Samsung: QA-Board (2020)



Source: https://samsung.github.io/qaboard/blog/2020/06/24/flame-graphs

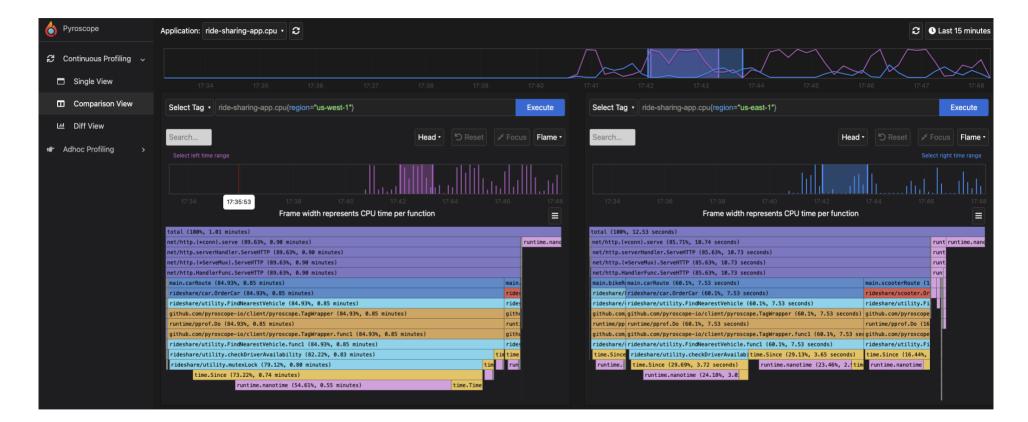
Microsoft Visual Studio: vscode-js-profile-flame (for JavaScript; 2020)

c: > Users > Connor > Documents > Git	hub > vscode-js	-debug > ≡ vscode-profile-2020-0	8-14-10-2	27-18.c	cpuprofile				
									Aa 💒 📅 🔗
1,760ms		3,520ms	5,28	30ms	7,040ms		8,800ms	10,600ms	12,300ms
processTicksAndRejections				on	StreamRead		processTimers	pro_; toD (program)	
runMicrotasks				(a	anonymous)		listOnTimeout runNextTic	k (; g bla	
(anonymous)	toDap	proces proce launc res (an	(a	re	eadableAddChunk		runNextTic runMicrota	s p g isS	
_activateByEvent	onced	proces ge (a doLau fin to	(a	ad	IdChunk		runMicrot; (anonymous)	g g iss	
activateByEvent	(anonymous)) (anony ge pr resol (an ti	(a	em	nit		process (anonymous)) <mark>g r (</mark> ga	
_activateExtensions	onced	l (anony ge ge resol exe	10	on	ndata	(anonyi t	process sendPending	e e i i i	
_activateExtensions	(anony	_send ru gi resol spa	ac	(a	anonymous)	handle ((anony p get		
(anonymous)	fromRu	send ex ex findI spa	CI	wr	riteOrBuffer	accet a	(anon) toKey	SS	
_activateExtension	fromRi	(anon _e _e (anon spa	C1	dol	DWrite	(anoi	_send isCaseInser	rssi	
actualActivateExtension	Stackl	write sp sp execS	CC I	(a	anonymous)	(a	send (anonymous)		
_activateExtension	rawLor	dowri st st sbamı	(;	(a	anonymous)	_1	(anon:	S	
_doActivateExtension	rawLo	(anon (; (; spawr	a	tr	ransform		write		
_loadCommonJSModule	get	(anon si si spawr		pu	JSh		doWri		
S	wai	write		(a	anonymous)		(anon		
require	(ar	aft((a	anonymous)		(anon		
(anonymous)	se1	onw)		re	eadableAddChunk	111 11	write		
(anonymous)	Tit	nex'		ad	idChunk		afte		
(anonymous)	in:	Tic		em	nit		onwr		
(anonymous)	em:	emi		(a	anonymous)		next		
(anonymous)	em:	emi		fi	ire		Tick		
(anonymous)	in:			(a	anonymous)		emit		
(anonymous)				_0	onMessage		emit		

Left Heavy view

Source: https://marketplace.visualstudio.com/items?itemName=ms-vscode.vscode-js-profile-flame

Pyroscope: flame graph (2020)



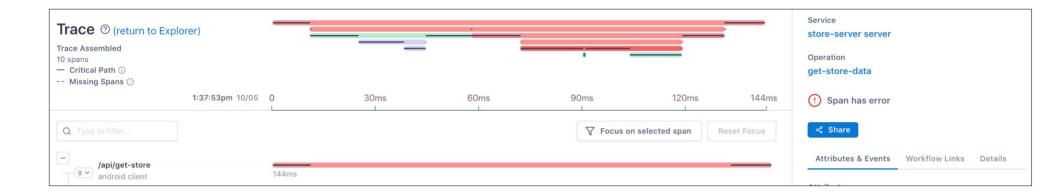
Source: https://pyroscope.io/blog/what-is-a-flamegraph/

Uber: pprof++ (2021; for Golang)

root															
ssa.(*Prog	gram).Build.fu	nc1													
ssa.(*Pack	kage).Build														
sync.(*On	ce).Do														
sync.(*On	ce).doSlow														
ssa.(*Pack	kage).build														
ssa.(*Fu.	ssa.(*Pr	ssa.(*builder).	assign	ssa.	(*builder).bu	ildF	uncDecl								
ssa.bui	ssa.(*Pr	ssa.(*builder	4	ssa.	(*builder).bu	ildF	unction								
	ssa.(*	ssa.(ssa.(*Funct	ti	ssa.(*builder).stmt								
	ssa				ssa.bui		ssa.(*builder).stmtLis	t							
					run		ssa.(*builder).stmt								
						I	ssa.(*builder).ass	ssa.(*bui	ssa.(*builde	ssa.(*builder)	ssa.(ssa	.(*builder).stmt	ssa
						1	ssa.(*bui		ssa	ssa.(*buil	ssa.(*builder	ssa		ssa.(*builder).stmtL	
						1	ssa.(*bu			ssa.(ssa.(*builder	ssa		ssa.(*builder).stmt	
							ssa.(*			ssa	ssa.(*builder	ssa		ssa.(
				1										ssa	
				- î		1							Ĩ.I		
				1				nni					11		
								11	111				ΪÏ		111
								11	111				11		1.1
								1							1
		11								· · · · ·			1		
		1							1.1						
		1													
								1				1			
													1		
														1	
	1														
							1								

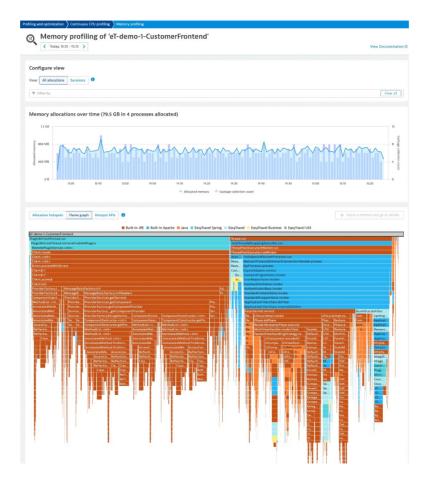
Source: https://www.uber.com/en-AU/blog/pprof-go-profiler/ (Pengfei Su)

Lightstep: flame graph (2021)



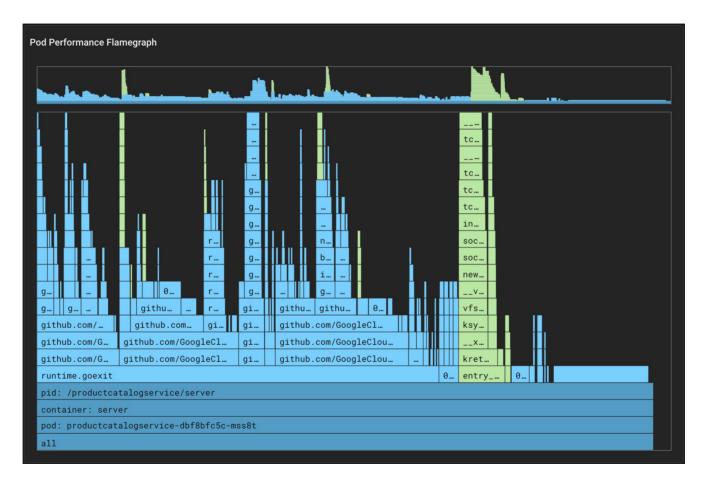
Source: https://www.instana.com/blog/instana-announces-the-industrys-first-commercial-continuous-production-profiler/

Dynatrace: allocation flame graph (2021)



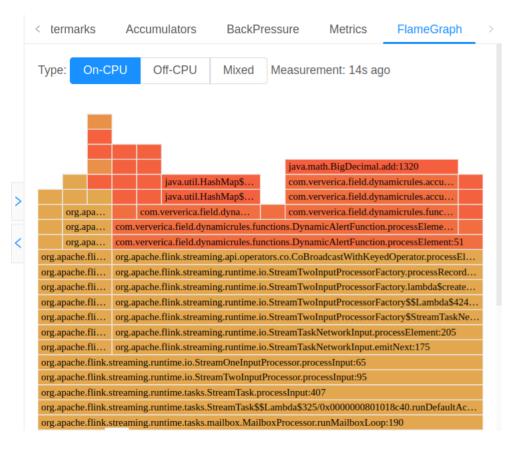
Source: https://www.dynatrace.com/support/help/how-to-use-dynatrace/diagnostics/memory-profiling

Pixie Labs: pod performance flamegraph (2021)



Source: https://docs.pixielabs.ai/tutorials/pixie-101/profiler/

Apache Flink: flame graphs (2021)



	java.lang.Thread.sleep:-2	java.lang.Thread.sleep:-2
	$com.ververica.field.dynamic rules.functions.Dynamic Alert Function.\ldots$	$com.ververica.field.dynamicrules.f\ldots$
	$com.ververica.field.dynamic rules.functions.Dynamic Alert Function.\ldots\\$	$com.ververica.field.dynamicrules.f\ldots$
	com.ververica.field.dynamic rules.functions.Dynamic Alert Function.product and the second s	cessElement:51
	org. a pache. flink. streaming. a pi. operators. co. CoBroadcast With Keyed Operators and the stream of the stre	erator.processElement1:125
	org.apache.flink.streaming.runtime.io.StreamTwoInputProcessorFactor	y.processRecord1:200
	org.apache.flink.streaming.runtime.io.StreamTwoInputProcessorFactor	y.lambda\$create\$0:165
	org.apache.flink.streaming.runtime.io.StreamTwoInputProcessorFactor	y\$\$Lambda\$424/0x00000008010b9
	org. a pache. flink. streaming. runtime. io. Stream Two Input Processor Factor and the stream of t	y\$StreamTaskNetworkOutput.emitR
ternal.misc.Unsafe.park:-2	org. a pache. flink. streaming. runtime. io. Stream Task Network Input. procession of the stream test of t	sElement:205
itil.concurrent.locks.LockSupport.parkNanos:252	org. a pache. flink. streaming. runtime. io. Stream Task Network Input. emit Network Input. Stream Task Network Input. Stream T	ext:175
itil.concurrent.locks.AbstractQueuedSynchroniz	org.apache.flink.streaming.runtime.io.StreamOneInputProcessor.proces	sInput:65
pache.flink.streaming.runtime.tasks.mailbox.Tas	org.apache.flink.streaming.runtime.io.StreamTwoInputProcessor.proces	ssInput:95
pache.flink.streaming.runtime.tasks.mailbox.Mai	org.apache.flink.streaming.runtime.tasks.StreamTask.processInput:407	
pache.flink.streaming.runtime.tasks.mailbox.Mai	org.apache.flink.streaming.runtime.tasks.StreamTask\$\$Lambda\$325/0	x0000000801018c40.runDefaultActi
oache.flink.streaming.runtime.tasks.mailbox.Mai	org.apache.flink.streaming.runtime.tasks.mailbox.MailboxProcessor.ru	nMailboxLoop:190

off-CPU

Source: https://nightlies.apache.org/flink/flink-docs-master/docs/ops/debugging/flame_graphs/

org.aj

org.ap

Embrace: Application-Not-Responding flame graph (2021)

NR PACKAGES	IL					ANR Duration	Group By First Sample	Time Filter	Version ays	10.8.3 😮 10.8.2 😵
App version 10.8.4, 10.8.3, 10.8.2 Fi	er									
50 1	10 150	200	250	300	350	400	450	500	550	600
na principal de la constitución de			and the second	- in the second party of the				and the second secon		illined and all a
50 5	00 150	200	250	300	350	400	450	500	550	600
om.android.internal.os.ZygoteInit.m	ain									
om.android.internal.os.RuntimeInit\$	@ethodAndArgsCaller.run									com.androCaller.ru
ava.lang.reflect.Method.invoke										java.langhod.invok
ndroid.app.ActivityThread.main										android.ahread.mai
ndroid.os.Looper.loop										android.oooper.loo
ndroid.os.Handler.dispatchMessage										android.otchMessag
ndroid.app.ActivityThread\$H.handleb	essage								android.oeCallbac	androidleMessag
ndroid.app.servertransaction.Transa						app.AcyThreadwrap			io.re…le.ru	an1
ndroid.app.servertransaction.Trnsa	ctionExecutor.executeCal	lback android.app.servertra	nsaction.T.onExecut	tor.executeLifecycleStat	android.a	app.AceLaunchActivi	t android Activit	andvit a	io.rr.ru	ant
ndroid.app.servertransaction.Launch				androiexecut andut		app.AcmLaunchActivi			io.rorma	ant
ndroid.app.ActivityThread.handleLau	-			androictivit andit		ivityOnCreat	android Activit		comNex	ant
ndroid.app.ActivityThread.performLa	-	android.apuseActivit		androictivit and ne		ppperformCreat	androidIfNeede		comNex	ant
ndroid.app.Instrumentation.callActi	rityOnCreate	android.apityIfNeede		androimResum andto		<pre>appperformCreat</pre>	androidyOnPaus		comade	cot
ndroid.app.Activity.performCreate		android.apvityOnPaus		androinResum andop		i.hvity.onCreat	androidormPaus		comade	
ndroid.app.Activity.performCreate		android.aperformPaus		com.apnResum andto		ie comlat	com.apponPaus		comtat	
om.app.ui.home.HomeActivity.onCreat		com.app.uity.onPaus		com.apnResum comto	com.aMod		. com.apponPaus		comde	
om.app.ui…eViewMode com.appinfl om.app.u.eViewMode com.appinfl		com.app.acty.onPaus androidx.fty.onPaus		androinResum andto		u andlat	android on Paus		comyI	
om.app.ueViewMode com.appinfl ndroidxy.getValu androidinfl		android.apty.onPaus		androi Kesun and to		anddre	androidyPause	a	comgl	
ndroidxy.getValu androidinfla		androiPause andse		andume ae ae	andrr.ge		cose		comer	
ndroidxy.getvalu androidinria ndroidxovider.ge androihildre		androiFause andse	androiction	5	comreat		cose		ander	
ndroidxvider.ge androiInflat		comuse	androixecut		cominit		. cou		andor	
om.app.u.ry.creat androi.hildre		comuse	androigethe		ct	anddre			ander	
om.app.u.el. <init androi="" inflat<="" td=""><td></td><td>comocu</td><td>androidStat</td><td></td><td></td><td>andlat</td><td></td><td></td><td>and_it</td><td></td></init>		comocu	androidStat			andlat			and_it	
ominit androihildre		comas	and Vie a.e			a a			and_it	
androiInflat		Comas	andVie ae			a a			andar	
andromTa ae		comJo	coe ae						andio	

Source: https://blog.embrace.io/solve-anrs-with-flame-graphs/

Polar Signals: parca Continuous Profiling (2021)

Download pprof	parch nodes	Reset View	Table I	Both	Icicle Graph
	Better Worse Worse				
root					
github.com/parca-dev/parca/pkg/s	github.com/polarsignals/frostdb.(*Table).writeBlock				g
github.com/parca-dev/parca/pkg/p	github.com/polarsignals/frostdb.(*TableBlock).Persist				
github.com/parca-dev/parca/pkg/p	github.com/polarsignals/frostdb.(*TableBlock).Serialize				
github.com/parca-dev/parca/pkg/	github.com/segmentio/parquet-go.(*mergedRowGroupRows).ReadRows				
github.com/polarsignals/frostdt	github.com/segmentio/parquet-go.(*mergedRowGroupRows).init				
github.com/polarsignals/frostdt	github.com/segmet github.com/segmentio/parquet-go.(*rowGroupRows).init				
github.com/polarsignals/frostdt	github.com/segm github.com/polarsignals/frostdb/dynparquet.(*remappedColumnChunk).Pa				
github.com/polarsignals/frostdt	github.com/segm github.com/segmentio/parquet-go.(*fileColumnChunk).Pages				
github.com/polarsignals/f gith	github.com/segm github.com/segmentio/parquet-go.(*filePages).init g				
github.com/polarsignals/f gith	github.com/segn bufio.NewReaderSize g				
github.com/segmentio/pai gith	github.com/pola github.com/segmentio/parquet-go.acquireReadBuffer				
github.com gi : gi' gil bufi	github.com/segn				
github.com y gil byte	githu gith git				
github.com ! gi byte	githu git				
github.com ! byte	gith gi				
github.com !	git gi				
github.com !	· git gi				
github.c 1	' git				
github.c 1	y git				
github.c	git git				
githut 1	· git				
refle	git				
refle					

Source: https://www.polarsignals.com/blog/posts/2022/08/30/optimizing-with-continuous-profiling/

Dockyard: Flame On (for Elixir apps; 2022)

Phoen	ix Live[Dashboo	ard			
Home	OS Data Enable	Metrics	Request Logger	Applicatio	ns Processes	Ports Sc
Module cowboy_ha	Indler		Function execute		Arit	у
cowboy_ham E: Elixir.	Phoenix.Endp	-0-/5 riginal:execut oint.Cowboy2H eb.Endpoint:c	andler:init/4			
Elixir El El:	.Flamegraphs		the second se	1/2	Elixir.Phoenix.Ro lists:foreach/2	uter:call/ Elixir.Fla
El	lixir.Plug.Co lixir.Enum:-r lixir.Plug.Co lixir.Plug.Co	educe/3-1			telemetry:-execut Elixir.Logg log Elixir.Phoe El El : E Elix; E	ge E Elixir.H
I	Sli: Eli F	Eli Elix			E EI E	Elixir. Elixir Elixir Elixir
						I Eli; Eli Eli

Source: https://dockyard.com/blog/2022/02/22/profiling-elixir-applications-with-flame-graphs-and-flame-on (Mike Binns)

OpenResty: Xray (2022)



Source: https://openresty.com/en/xray (Yichun Zhang)

Elastic: universal profiling (2022)

😔 elastic	$\ensuremath{\mathbb{Q}}$ $\ensuremath{\mbox{Find}}$ apps, content, and more. Ex: Discover	×/	© & @
D Observability Unive	ersal Profiling Flamegraphs Flamegraph		
d Observability	Universal Profiling		
Overview Alerts Cases	Flamegraph Differential flamegraph		
Logs		(iii) → Sep 20, 2022 @ 04:00:00.000 → Sep 20, 2022 @ 06:00:	00.000 C Refresh
Stream			
Anomalies		\bigcirc ×	Show information window
Categories			
Infrastructure	root: Represents 100% of CPU time. Python: _handle_and_close_when_done() in baseserver.py #34 Python: handle() in gevent.py #119 Python: handle() in gevent.py #119		
Inventory	Python: handle() in base_async.py #55 Python: handle_request() in ggevent.py #127		
Metrics Explorer	Python: handle_request() in base_async.py #108 Python: sentry_patched_wsgi_app() in flask.py #80 Python: _call_() in wsgi.py #122 Python: _lambdas() in flask.py #85		
АРМ	Python:call() in app.py #2086 Python: wrapper() in helpers.py #20		
Services	Python: traced_wsgi_app() in patch.py #255 Python: wsgi_app() in app.py #2073		
Traces	Python: full_dispatch_request() in app.py #1516 Python: wrapper() in helpers.py #20		
Dependencies	Python: _traced_request() in patch.py #458 Python: dispatch_request() in app.py #1480		
Service Map	Python: trace_func() in wrappprs.py #19 Python: _wrapper() in metric_collector.py #403 Python: wrapper() in flask_wauth_client.py #271	Python: put_media_key() in	and the second state
Uptime	Python: _wrapper() in decorators.py #61 Python: submit_segments() in segment_retention.py #93	video_key_manager.py #46	
Monitors	Python: get_latest_video() in video_manager.py #578 Python: put_ Python: query() in video_manager.py #77 Python: do_t		
TLS Certificates	< > Search string	Exclusive CPU: 0.00% Samples: 12370	

Source: https://www.elastic.co/observability/universal-profiling

... and more

0		🕧 Pulls Issues Codespaces Marketplace Explore 🧔 🕹 🗸 🖓	-				
flamegraph OR "fl	lame graph"	Search					
Repositories	407	O Connect from an allowed IP address to see search results within the intel-innersource organization.					
Code	29K	O Connect from an allowed IP address to see search results within the intel-sandbox organization.					
Commits	72K+	Single sign-on to see search results for organizations within the Intel enterprise.					
lssues	19K	407 repository results Sort: Best match •					
Discussions	158	407 repository results					
Packages	5	💂 brendangregg/ FlameGraph					
Marketplace	0	Stack trace visualizer ☆ 13.9k ● Perl Updated on Oct 4					
Topics	6						
Wikis	413	(Dec 2022))				

Thanks for all the open source contributions!