# **Chromium on Android** Web Performance

How we **doubled Chromium's Speedometer** scores & developed the **LoadLine** page load benchmark

eseckler@chromium.org | Eric Seckler | Google UK | Android Web Perf

February, 2025

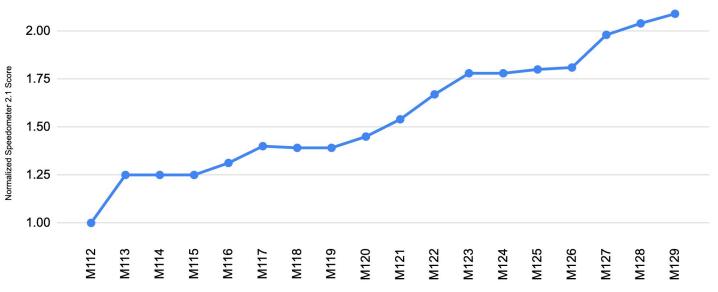






Speedometer 2.1 scores on Android Chrome increased 109% (e.g. Pixel Tablet: 97 in 2023 to 203 today).

#### Speedometer uplift on Android



Chrome Release

#### Faster Speedometer => Faster page load

549 🔿					0	547 0						NO.		
New tab	X @ https://docs.nocole.com/dis_X					Nov	v tab		0	< e> http://d+				
$\diamond \leftrightarrow \times \diamond$	docs.google.com/document/d/1-BCM2KW90WUllsgUjXRrqYQ0dbCHoDytI7_JfeuHYFFw/edit	\$	±	1	в	۵ (	€ )	» >	× 0	docs.google.com/document/d/1-8CM2KW9OWUlsgUjXRrqYQOdbCHoDytl7_JfeuHYFFwledit	☆	±	<	0:

#### M112 - 00;00

M129 - 00;00

# Thank you for your contributions!

- Chrome/V8
- Android
- Pixel

- ARM
- Qualcomm

The browser built to be 🕜 fast Google nding far 5.300+ lor Google

# Table of contents

**O2** Deep-dive: Build optimizations

**03** Tooling to enable further analysis

**04** Beyond Speedometer: LoadLine

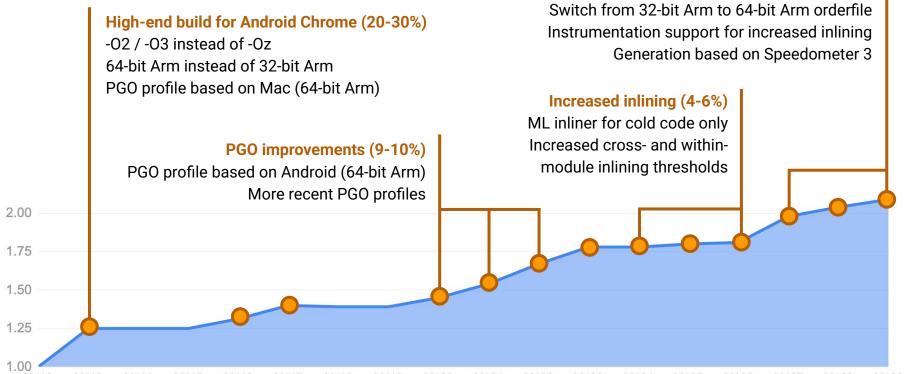
**05** Q&A

#### Breaking down the timeline!

- 01.1 Build improvements
- 01.2 V8 and Blink improvements
- 01.3 Scheduling & OS



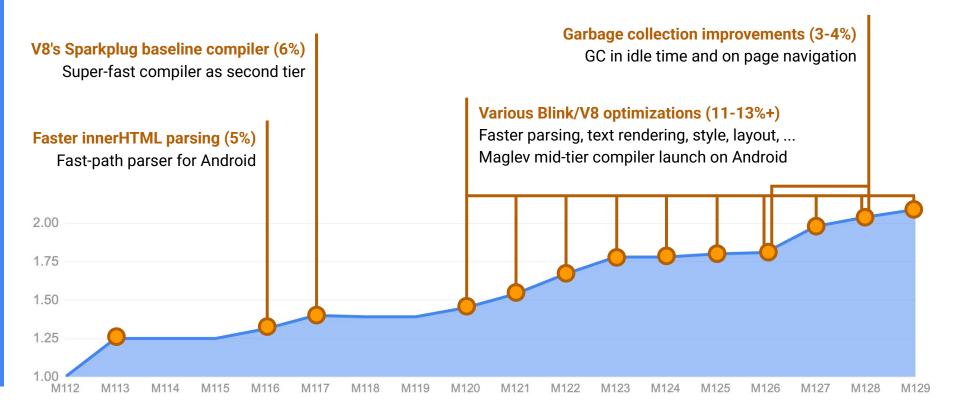
#### **Build improvements**



**Orderfile improvements (4-7%)** 

M112 M113 M114 M115 M116 M117 M118 M119 M120 M121 M122 M123 M124 M125 M126 M127 M128 M129

#### V8 and Blink improvements



#### Scheduling & OS

#### Higher prioritization of CrRendererMain (4%)

Flagging CrRendererMain as high-priority thread via Android Dynamic Performance Framework (ADPF)



# Insights: Build improvements Profiling to understand Chrome's CPU bottlenecks

## PMU data: high stalls & frontend-bound workload

Suite	Frontend stalls	Backend stalls
Angular2-TypeScript-TodoMVC	45.92%	18.32%
AngularJS-TodoMVC	43.12%	17.47%
BackboneJS-TodoMVC	55.49%	14.97%
Elm-TodoMVC	28.91%	19.35%
EmberJS-Debug-TodoMVC	28.72%	24.77%
EmberJS-TodoMVC	40.41%	20.01%
Flight-TodoMVC	52.32%	17.76%
Inferno-TodoMVC	44.32%	14.69%
Preact-TodoMVC	44.24%	19.10%
React-Redux-TodoMVC	28.03%	16.74%
React-TodoMVC	40.68%	16.36%
Vanilla-ES2015-Babel-Webpack-TodoMVC	38.50%	17.24%
Vanilla-ES2015-TodoMVC	38.42%	17.99%
VanillaJS-TodoMVC	40.66%	17.52%
VueJS-TodoMVC	44.08%	17.19%
jQuery-TodoMVC	28.41%	20.79%
Overall	36.22%	19.27%

Speedometer has **many branches** (~20% of instructions)

Branch mispredicts are costly in many ARM CPUs (instruction cache prefetch)

**Optimizing code/branch layout** for branch target buffer, caches, and CPU frontend parallelism is critical

CPUs with larger branch predictors and instruction caches are beneficial

## Optimized PGO (& CPU) -- bottleneck moves to backend

Subtest	IPC	Frontend stall rate	Backend stall rate
Charts-chartjs	3.19	16.70%	36.16%
Charts-observable-plot	3.01	15.28%	40.54%
Editor-CodeMirror	2.86	23.91%	28.93%
Editor-TipTap	3.36	11.77%	35.44%
NewsSite-Next	2.15	28.81%	35.70%
NewsSite-Nuxt	2.18	28.54%	35.41%
Perf-Dashboard	2.23	29.63%	35.55%
React-Stockcharts-SVG	2.68	20.55%	38.79%
TodoMVC-Angular-Complex-DOM	2.38	24.72%	37.73%
TodoMVC-Backbone	2.10	31.98%	33.02%
TodoMVC-JavaScript-ES5	3.11	15.26%	42.27%
TodoMVC-JavaScript-ES6-[]	2.96	13.74%	47.57%
TodoMVC-jQuery	3.09	14.26%	42.62%
TodoMVC-Lit-Complex-DOM	2.16	22.96%	43.60%
TodoMVC-Preact-Complex-DOM	1.88	22.73%	47.52%
TodoMVC-React-Complex-DOM	2.62	22.74%	37.56%
TodoMVC-React-Redux	2.91	24.52%	32.13%
TodoMVC-Svelte-Complex-DOM	1.91	24.24%	46.19%
TodoMVC-Vue	2.30	23.91%	39.81%
TodoMVC-WebComponents	2.14	23.08%	43.89%
Overall	2.69	20.78%	38.50%

PGO increases portion of **not-taken** branches by placing hot blocks into fall-through paths

Not-taken branches **consume no BTB space** and enable more efficient utilization of caches and frontend width

**Orderfile** (function ordering) improves on top of this by **reducing iTLB misses** 

Backend bottlenecks are now focus of our investigations

Pixel 9 Pro (Cortex X4), Chrome M128, Speedometer 3.0, CrRendererMain only

#### Stalls remain high overall -- high mem-boundedness

Subtest	Frontend stalls	FE L3+ stalls	Backend stalls	BE L3+ stalls
Charts-chartjs	16.70%	7.91%	36.16%	17.30%
Charts-observable-plot	15.28%	6.83%	40.54%	18.94%
Editor-CodeMirror	23.91%	10.78%	28.93%	15.93%
Editor-TipTap	11.77%	4.21%	35.44%	9.29%
NewsSite-Next	28.81%	12.35%	35.70%	16.24%
NewsSite-Nuxt	28.54%	11.57%	35.41%	17.00%
Perf-Dashboard	29.63%	14.39%	35.55%	18.30%
React-Stockcharts-SVG	20.55%	8.84%	38.79%	19.20%
TodoMVC-Angular-Complex-DOM	24.72%	9.13%	37.73%	19.51%
TodoMVC-Backbone	31.98%	11.93%	33.02%	16.90%
TodoMVC-JavaScript-ES5	15.26%	6.82%	42.27%	14.52%
TodoMVC-JavaScript-ES6-[]	13.74%	7.17%	47.57%	18.90%
TodoMVC-jQuery	14.26%	6.49%	42.62%	14.59%
TodoMVC-Lit-Complex-DOM	22.96%	10.56%	43.60%	21.67%
TodoMVC-Preact-Complex-DOM	22.73%	11.84%	47.52%	28.29%
TodoMVC-React-Complex-DOM	22.74%	8.54%	37.56%	20.13%
TodoMVC-React-Redux	24.52%	9.04%	32.13%	15.17%
TodoMVC-Svelte-Complex-DOM	24.24%	13.58%	46.19%	26.11%
TodoMVC-Vue	23.91%	9.46%	39.81%	21.19%
TodoMVC-WebComponents	23.08%	10.44%	43.89%	19.78%
Overall	20.78%	8.93%	38.50%	16.96%

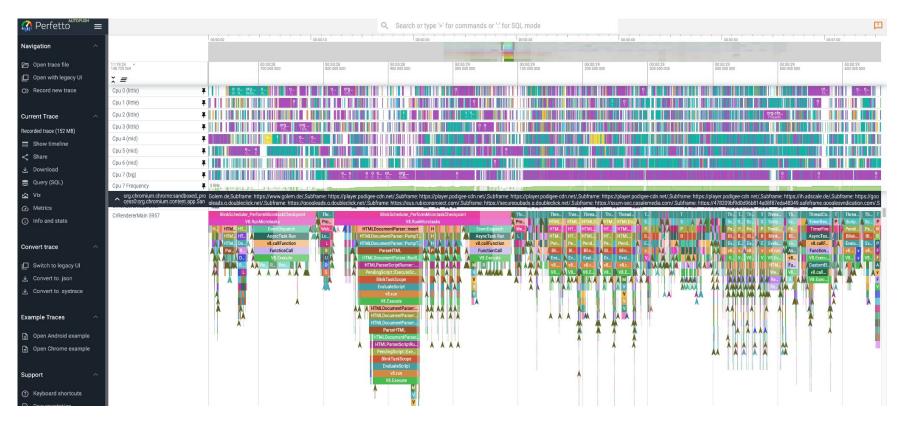
Just under half of the stalls on Cortex X4 stem from **L2 misses** 

Plan to attribute L2 misses to code and data via profiling

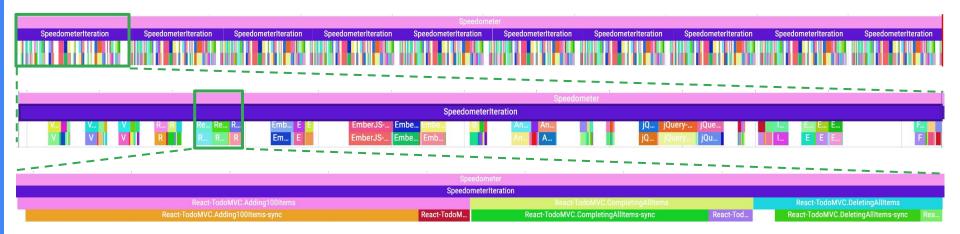
Pixel 9 Pro (Cortex X4), Chrome M128, Speedometer 3.0, CrRendererMain only

# Tooling to enable further analysis

#### Perfetto with Chrome and system data sources



#### Breaking down Speedometer execution in traces



Annotate Perfetto traces with Speedometer phases + perf/simpleperf integration + C++/V8 symbolization

Allows breakdowns of callstacks, PMU counters, etc. by subtest

#### AUTOPUSH Perfetto \$ e.g. select \* from sched left join thread using(utid) limit 10 1 = 00:00:00 00:00:20 00:00:40 00:00:50 00:01:00 00:01:10 00-00-10 00-00-30 Navigation 🗁 Open trace file 18:19:03 + 236 604 230 00:00:30 000 000 000 191ms 926us 478ns ID Open with legacy UI ž = ~ # [ O) Record new trace Cpu 8 cpu-cycles ~ 🖡 🚺 Cpu 8 instructions **Current Trace** IPC × ~ 🖡 🚺 speedometer3\_100000\_samples\_wit × ∓[ Speedometer phase Ρ h\_v8\_many\_categories.zip (37 MB) Show timeline ∧ Renderer 7862 Speedometer 3 Results Summary, Subframe: https://chromium-workloads.web.app/ Share CrRendererMain Callstacks 7888 Process Callstacks 🛃 Download ocessService0:3 7862 main thread Query (SQL) AnimationFrame 🖄 Viz Metrics CrRendererMain 7888 Info and stats CrRendererMain 7888 Th... T Pr... AP Pr., Pr., P P... Pro... Convert trace Pa... Switch to legacy UI .↓ Convert to .json V.... v8... ۷... F Fu... E. F F.,. Fu... Current Selection Standalone Query (1) × **↑** ∨ Example Traces Area Selection Flamegraph Selection Pivot Table Open Android example Perf Samples ~ Add filter.. Open Chrome example v8::internal::Execution::Call(v8::internal::Isolate\*, v8::internal::Handle<v8::internal::Object>, v8::internal::Handle<v8::internal::Object>, int, v8::internal::Handle<v8::internal:: b b b Builtins\_JSEntry Builtins\_JSEntryTrampoline b а Support b v unknown runSvi b b b b ⑦ Keyboard shortcuts Suites.push.tests.i q b b Documentation b b Builtins\_CallApiCallbackGeneric B blink::Node::DispatchSimulatedClick(blink::Event const\*, blink::SimulatedClickCreationScope) b b FI Flags blink::HTMLElement::click() b b Report a bug b b blink::(anonymous namespace)::v8\_html\_element::ClickOperationCallback(v8::FunctionCallbackInfo<v8::Value> const&)

Y metric: samples 👻 Add pivot or filter... 9

68,745 (100.00%)																	
TodoMVC-jQuery	NewsSite-Next	NewsSite-Nuxt	React-Stockcha	Editor-TipTap	TodoMVC-J	TodoMVC-J	Charts-c	Charts	TodoM	TodoM	Perf-Da	TodoM	Todo	Tod	Tod 1	Tod Ed	li T 1
unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unknown	unkno	unkno	unkno	unk	unk u	ink un	u ı
art_quick_generic_jni_trampoline	art_quick_generic_jni_tr	art_quick_generi	art_quick_gener	art_quick_gen	art_quick_g	art_quick_g	art_quick	art_qui	art_qui	art_qui	art_qui	art_qui	art_qu	art	art a	art ar	t a a
Java_J_N_M1Y_1XVCN	Java_J_N_M1Y_1XVCN	Java_J_N_M1Y	Java_J_N_M1Y	Java_J_N_M1	Java_J_N	Java_J_N	Java_J_N	Java_J	Java_J	Java_J	Java_J	Java_J	Java	Jav	Jav J	Jav Ja	J
*content::JNI_ContentMain_Start	content::JNI_ContentMa	content::JNI_Co	content::JNI_Co	content::JNI	content::JNI	content::JNI	content::	*content	content	conten	conten	conten	conte	cont	con c	con°co	c c
content::RunContentProcess	content::RunContentPro	content::RunCon	content::RunCo	content::RunC	content::Run	content::Ru	content::	content	content	conten	conten	conten	conte	cont	con c	con co	c o
ContentMainRunnerImpl::Run	ContentMainRunnerImp	ContentMainRun	ContentMainRu	ContentMain	ContentMai	ContentMai	Content	Content	Conten	Conten	Conten	Conten	Conte	Con	Con C	Co Co	o C (
content::RunOtherNamedProces	content::RunOtherName	content::RunOth	content::RunOt	content::RunO	content::Run	content::Ru	content::	content	content	conten	conten	conten	conte	cont	con c	con co	c (
content::RendererMain	content::RendererMain	content::Render	content::Render	content::Rend	content::Ren	content::Re	content::	content	content	conten	conten	conten	conte	cont	con c	con co	c c
RunLoop::Run	RunLoop::Run	RunLoop::Run	RunLoop::Run	RunLoop::Run	RunLoop::Run	RunLoop::R	RunLoop:	RunLoo	RunLoo	RunLo	RunLo	RunLo	RunLo	Run	Run F	Run Ru	I R F
ThreadControllerWithMessagePu	ThreadControllerWithM	ThreadControlle	ThreadControll	ThreadContro	ThreadContr	ThreadCont	ThreadC	Thread	Thread	Thread	Thread	Thread	Threa	Thr	Thr 1	Thr Th	n T 1
MessagePumpDefault::Run	MessagePumpDefault::	MessagePumpD	MessagePump	MessagePum	MessagePu	MessagePu	Message	Messa	Messa	Messa	Messa	Messa	Messa	Mes	Me M	Me M	e M. M
non-virtual thunk to base::sequen	non-virtual thunk to bas	non-virtual thunk	non-virtual thun	non-virtual th	non-virtual t	non-virtual t	non-virtu	non-virt	non-virt	non-vir	non-vir	non-vir	non-vi	non	non n	non no	n r
*ThreadControllerWithMessagePu	*ThreadControllerWithM	*ThreadControlle	*ThreadControll	*ThreadContro	*ThreadContr	*ThreadCont	*ThreadC	*Thread	*Thread	*Thread	*Thread	*Thread	*Threa	°Thr	Thr 1	Thr <sup>*</sup> Th	n °T*1
*ThreadControllerWithMessagePu	*ThreadControllerWithM	*ThreadControlle	*ThreadControll	*ThreadContro	*ThreadContr	ThreadCont	*ThreadC	*Thread	*Thread	*Thread	*Thread	*Thread	Threa	Thr	Thr T	Thr <sup>°</sup> Th	ı °T°1
void base::TaskAnnotator::RunTa	void base::TaskAnnotat	void base::Task	void base::Task	void base::Ta	void base::T	void base::T	void bas	void ba	void ba	void ba	void ba	void ba	void b	void	voi v	voi <sup>°</sup> vo	i °v °۱
*TaskAnnotator::RunTaskImpl	TaskAnnotator::RunTas	*TaskAnnotator::	*TaskAnnotator::	TaskAnnotato	TaskAnnota	TaskAnnota	TaskAnn	<sup>*</sup> TaskAn	<sup>*</sup> TaskAn	*TaskAn	TaskAn	TaskA	<sup>°</sup> TaskA	Tas	Tas T	Tas <sup>°</sup> Ta	<sup>*</sup> T*1
*base::OnceCallback <void ()="">::Run</void>	base::OnceCallback <voi< td=""><td><sup>•</sup>base::OnceCallb</td><td>base::OnceCall</td><td>base::OnceCa</td><td><sup>*</sup>base::Once</td><td>base::Once</td><td><sup>•</sup>base::On</td><td>base::0</td><td><sup>*</sup>base::0</td><td>base::</td><td>base::0</td><td><sup>•</sup>base::</td><td>base::</td><td>bas</td><td>bas b</td><td>bas <sup>°</sup>ba</td><td> b t</td></voi<>	<sup>•</sup> base::OnceCallb	base::OnceCall	base::OnceCa	<sup>*</sup> base::Once	base::Once	<sup>•</sup> base::On	base::0	<sup>*</sup> base::0	base::	base::0	<sup>•</sup> base::	base::	bas	bas b	bas <sup>°</sup> ba	b t
ProxyMain::BeginMainFrame	ProxyMain::BeginM	ProxyMain::Be	ProxyMain:: T.	ProxyMain::B	ProxyMain::	ProxyMain::	Proxy T	Proxy	Proxy	Proxy	Proxy	Proxy	Prox	Pro	Pro F	Pro P.	. P. F
*LayerTreeHost::BeginMainFrame	LayerTreeHost::Be	LayerTreeH L	LayerTreeH D.	LayerTreeHo	LayerTree L	LayerTree L	Layer D	Lay L.	*Laye L	Layer	Layer	Laye L	*Laye	*L L	L L L	L. *L	. L*
WidgetBase::BeginMainFrame	WidgetBase::Begin	WidgetBas L	WidgetBas S.	WidgetBase::	WidgetBa L	WidgetBa L	Widg S	Wid L.	Widg L	Widg	Widge	Wid L	Wid	W L	W L V	W. L. W	. L
WebFrameWidgetImpl::BeginM*	WebFrameWidgetI	WebFrame n	WebFrame *V.	WebFrameW	WebFram r	WebFram r	WebF V	Web n.	Web r	Web	WebFr	Web r	Web	W *r	W n V	W. n. W	**r *
Page::Animate	Page::Animate	Page::Anim W	Page::Anim V.	Page::Animate	Page::Ani V	Page::Ani V	Page: V	Pag W	Page V	Page	Page::	Pag N	Pag	°P °V	P V F	P W P	°V° °
PageAnimator::ServiceScripted	PageAnimator::Ser *	PageAnim W	PageAnim b.	PageAnimat	PageAni V	PageAni V	Page b	Pag W	Page V	Page	Page	Pag \	Pag	P V	P V F	P W P	. V
*PageAnimator::ServiceScripted*	PageAnimator::Ser	PageAnim P	PageAnim b.	PageAnimat	PageAni F	PageAni F	Page b	Pag P.	Page F	Page	Page	Pag F	Pag	°P °F	P F F	P P P	* °F* *
operator()<(lambda at//thir	operator()<(lambd	operator()< P	operator()< V.	operator()<(I	operator() F	operator() F	opera V	oper P.	oper F	oper	operat	oper F	oper	°o 🖡	o F c	o P °o	. *F* *
PageAnimator::ServiceScripted	PageAnimator::Ser	PageAnim L	PageAnim F.	PageAnimat	PageAni L	PageAni L	Page F	Pag L.	Page L	Page	Page	Pag L	Pag	P L	P L F	P L. °P	. L°
*ScriptedAnimationController::E	ScriptedAnimation	ScriptedAn L	ScriptedAn E.	ScriptedAni	ScriptedA L	ScriptedA L	Script E	Scri L.	<sup>°</sup> Scrip L	Scrip	Script	Scri L	*Scri		S L S	S L. <sup>*</sup> S	. L*
FrameRequestCallbackCollecti	FrameRequestCall	FrameRequ <sup>*</sup> L	FrameRequ	FrameReque	FrameRe	FrameRe 1	Fram v	Fra L.	Fra L	Fram	Frame	Fra L	Fra	Fr L	F L F	<b>°L</b> . F	. * °L *
V8FrameCallback::Invoke	V8FrameCallback:	V8FrameC L.	V8FrameC v	V8FrameCall	V8Frame	V8Frame	V8Fra v	V8F <sup>°</sup> L	V8Fr *	V8Fr *	V8Fra	V8Fr *	V8Fr	V *	٧* ١	/°L V.	** *
*V8FrameRequestCallback::Invo	V8FrameRequestC	V8FrameR L.	V8FrameR B.	*V8FrameReq	V8Frame	*V8Frame	<sup>•</sup> V8Fra B	<sup>°</sup> V8F L	<sup>°</sup> V8Fr	°V8Fr	<sup>°</sup> V8Fra	<sup>•</sup> V8Fr	<sup>•</sup> V8Fr	°V	۲ ۲.	/ L <sup>°</sup> V.	•
V8FrameRequestCallback::Invo	V8FrameRequestC	V8FrameR L.	V8FrameR B.	V8FrameReq	V8Frame	V8Frame	V8Fra B	V8F L	V8Fr	V8Fr	V8Fra	V8Fr	V8Fr	V	V \	/ L V.	
blink::bindings::CallbackInvoke	blink::bindings::Call	blink::bindi L.	blink::bindi u.	blink::binding	blink::bin	blink::bin	blink:: u	blin L	blink	blink:	blink:	blink	blink	bl	b b	L b	
blink::bindings::CallbackInvoke	blink::bindings::Call	blink::bindi L.	<sup>*</sup> blink::bindi x	blink::binding	blink::bin	<sup>•</sup> blink::bin <sup>•</sup>	blink: u	blin L	blink	blink:	blink::	blink	<sup>•</sup> blink	°bl *	b b	o°L °b	
V8ScriptRunner::CallFunction	V8ScriptRunner::C	V8ScriptRu L	V8ScriptRu e	V8ScriptRun	V8ScriptR	V8ScriptR	V8Scr I.	V8S	V8S	V8Sc	V8Scr	V8S	V8S	V	V \	/ V.	
Function::Call	Function: Micro	Functi M L	Function::C V	Function::Call	Function:	Function:	Funct	Fun **	Func	Func	Functi	Func	Fun	F	N F N	• F	
Execution::Call	Execution: Micro	Execu M L	Execution:: v	Execution::Call	Execution	Execution	Execu	Exe	Exec	Exec	Execu	Exec	Exec	E	NEN	E	

#### Additional low-level data sources: ETM and SPE

Query resu	ılt (10 rows) - 85.7m	S SE	ELECT	h.*, s.name F	ROM PERF_INSTRU	JCTION	RANGE h, frame f	USING (frame_id), symbol s	s ON (f.syn	nbol_set_id = s.id) WHERE start = 499699399144 LIMIT 10
id	type	cpu	utid	cycle_set_id	start	length	instruction_count	avg_cycles	frame_id	
278perf_:	instruction_range	7	4	276	499699399144	16	4	8.36363636363636363	1212	<pre>std::Cr::tree_node_base<void*>*&amp; std::Cr:: std::Cr::map_value_compare<char const*,="" std::_<br="">const*&gt;, true&gt;, std::Cr::allocator<std::cr:: const*&gt;(std::Cr::tree_end_node<std::cr::tree< pre=""></std::cr::tree<></std::cr:: </char></void*></pre>
280 perf_:	instruction_range	7	4	280	499699399144	16	4	13	1212	<pre>std::Cr::tree_node_base<void*>*&amp; std::Cr:: std::Cr::map_value_compare<char const*,="" std::<br="">const*&gt;, true&gt;, std::Cr::allocator<std::cr::v const*&gt;(std::Cr::tree_end_node<std::cr::tree< pre=""></std::cr::tree<></std::cr::v </char></void*></pre>
283 perf_:	instruction_range	7	4	283	499699399144	16	4	299	1212	<pre>std::Cr::tree_node_base<void*>*&amp; std::Cr:: std::Cr::map_value_compare<char const*,="" std::<br="">const*&gt;, true&gt;, std::Cr::allocator<std::cr:: const*&gt;(std::Cr::_tree_end_node<std::cr::tree_end_node< pre=""></std::cr::tree_end_node<></std::cr:: </char></void*></pre>
466 perf_:	instruction_range	7	4	466	499699399144	16	4	35	1212	<pre>std::Cr::tree_node_base<void*>*&amp; std::Cr::1 std::Cr::map_value_compare<char const*="" const*,="" std::="">, true&gt;, std::Cr::allocator<std::cr::y< pre=""></std::cr::y<></char></void*></pre>

Query result (100 rows) - 2.4ms INCLUDE PERFETTO MODULE linux.perf.spe; SELECT * FROM linux_perf_spe_record where ts >= 711265250880000 limit 100;											
ts	utid	exception_level	instruction_frame_id	operation	data_virtual_address	data_physical_address	total_latency	issue_latency	translation_latency	data_source	
711265250880075	2	EL0	4867	BRANCH	0	0	9	8	0	NULL	
711265250882109	2	EL0	75188	LOAD	481039752721	37495930385	14	8	1	L1D	
711265250888579	2	ELØ	73844	LOAD	481083917848	41576699416	2565	43	1499	DRAM	
711265250890898	2	EL0	74102	BRANCH	0	0	4	3	0	NULL	
711265250892852	2	ELØ	938	BRANCH	0	0	4	3	0	NULL	
711265250895415	2	ELØ	73967	LOAD	510653862160	45780164880	10	6	1	L1D	

# **Beyond Speedometer**

#### CUJs exercise browser components differently

...

Dreeses	Threed	Component		Benchmark / CUJ									
Process	Thread		JetStream	Speedometer	Page load	Scrolling	Tap/Typing						
Browser	UI				critical path	critical path							
	Network				critical path								
	ThreadPool												
Renderer	Main	V8	impacts score	impacts score	critical path								
		Blink		impacts score	critical path								
	Compositor				critical path	critical path	TBD						
	ThreadPool	GC/compile	impacts score	impacts score	critical path								
		Raster			critical path								
	JS workers	WASM	impacts score										
GPU					critical path	critical path							
SurfaceFlir	nger				critical path	critical path							

Degree of component usage (CPU-time-based)

#### CUJs exercise browser components differently

5		Component	Benchmark / CUJ									
Process	Thread		JetStream	Speedometer	Page load	Scrolling	Tap/Typing					
Browser	UI				critical path	critical path						
	Network				critical path							
	ThreadPool											
Renderer	Main	V8	impacts score	impacts score	critical path							
		Blink		impacts score	critical path							
	Compositor				critical path	critical path	TBD					
	ThreadPool	GC/compile	impacts score	impacts score	critical path							
		Raster			critical path							
	JS workers	WASM	impacts score									
GPU					critical path	critical path						
SurfaceFlir	nger				critical path	critical path						
						ing new Is in 2024/25	Google					



# Difficulties developing a page load benchmark

#### Relevance

Select ~5 representative sites based on **product needs** and **performance characteristics** 

Analyzed ~50 popular sites in 20+ dimensions via traces; clustering similar ones.

Strive for maximum coverage

#### **Metrics**

General-purpose loading metrics (LCP, FCP, ..) don't work well for low # sites

Custom instrumentation to enable **site-specific** metrics tracking readiness to interact

Strive for normal distribution

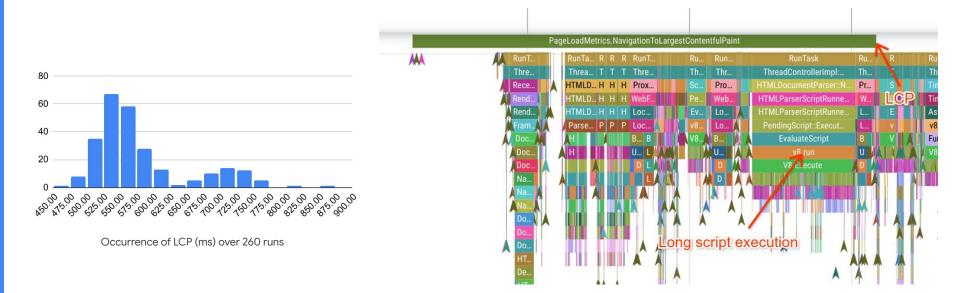
#### Noise

Sites evolve over time and behave differently in different geographies

Record and replay resource loads via **WPR on device**; avoid incompatible sites

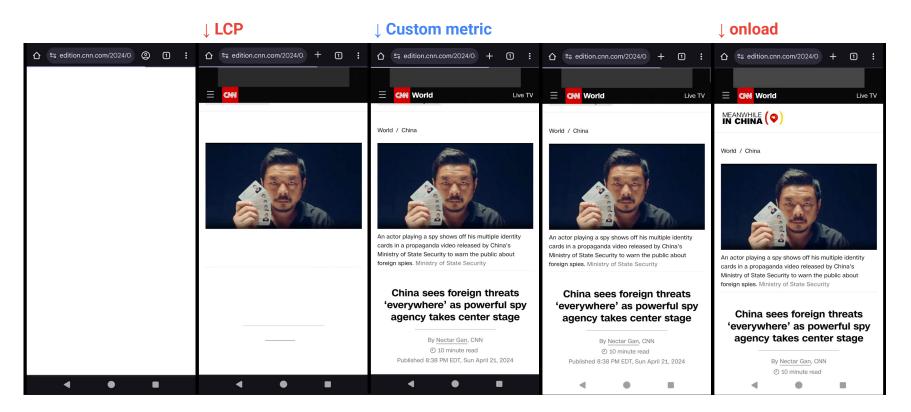
Detect 1% change in 1hr

#### Example: Metric bimodality



Bimodality caused by LCP-related paint running before or after a long unrelated script execution

### Example: Tracking a UX-relevant moment



Custom metrics aim to track the earliest moment that the main content is loaded and ready for interaction

### LoadLine stories: Phone (mobile) configuration

Page (mobile)	CUJ type / Product narrative	Performance characteristics	Metric
<u>amazon.co.uk</u> product page	Shopping: high usage	<ul> <li>average page load, large workload, large DOM/JS (but heavier on DOM)</li> <li>high on OOPIFs, input, http(s) resources, frame production</li> </ul>	JS ready
<u>cnn.com</u> article	News: high usage	<ul> <li>slow page load, large workload, large DOM/JS (but heavier on JS)</li> <li>high on iframes, main frame, local storage, cookies, http(s) resources</li> </ul>	Main content created
google.com search results	Search: largest single CUJ	<ul> <li>fast page load, average workload, average DOM + JS</li> <li>high on main frame, local storage, video</li> </ul>	LCP
<u>globo.com</u> homepage	News / web portal: high usage	<ul> <li>slow page load, large workload, small DOM, large JS</li> <li>high on iframes, OOPIFs, http(s) resources, frame production, cookies</li> </ul>	Cookie banner closed
wikipedia.org article	Reference work: simple/fast site	<ul> <li>fast page load, small workload, large DOM, small JS</li> <li>low on iframes, http(s) resources, frame production</li> </ul>	Last important event

### LoadLine stories: Tablet (desktop) configuration

Page (desktop)	CUJ type / Product narrative	Performance characteristics	Metric
amazon.co.uk product page	Shopping: high usage	<ul> <li>average page load, large workload, large DOM, average JS</li> <li>high on OOPIFs, http(s) resources, frame production</li> </ul>	JS ready
<u>cnn.com</u> article	News: high usage	<ul> <li>slow page load, large workload, large DOM/JS (but heavier on JS)</li> <li>high on iframes, local storage, video, frame production, cookies</li> </ul>	Main content created
google.com search results	Search: largest single CUJ	<ul> <li>fast page load, low workload, low DOM + JS</li> <li>high on main frame, local storage, low on video</li> </ul>	LCP
<u>youtube.com</u> video page	Media: high usage	<ul> <li>slow page load, very high workload, large DOM, small/average JS</li> <li>high on video</li> </ul>	Cookie banner closed
docs.google.com document	Productivity: expect increased relevance, challenging workload	<ul> <li>slow page load, large workload, large DOM + JS (heavier on JS)</li> <li>high on main frame, font resources</li> </ul>	LCP

Larger focus on productivity & challenging stories for the tablet configuration

#### Caveats

Today, LoadLine is an internal Chromium optimization target – not (yet) capable of comparing browsers or platforms.

- Built for Android site selection primarily based on mobile browsing (and Chromium only)
  - We provide phone (mobile) and tablet (desktop) workloads
- Covers fundamental CPU/GPU browser performance, but doesn't cover many networking intrinsics
  - May be extended with e.g. a traffic-shaping proxy to approximate some networking effects
- Not a micro-benchmark: Workload can change depending on device characteristics (e.g. frame rate)
  - Reflects the adaptable nature of the web and end-to-end page load performance
- Limited number of stories, some browser features are not captured
- Custom metrics are rudimentary today, so caution needed when evaluating browser behavior changes



#### Based on crossbench

- Setup <u>crossben.ch</u>
- ./cb.py loadline-phone

#### Available as v1.1 now

- Docs: <u>bit.ly/loadline</u>
- Feedback encouraged





# Thank you



Android