H

serverless performance

algolia

Stories Popularity All time

Serverless Performance: Cloudflare Workers, Lambda and LambdaEdge(https://blog.cloudflare.com/serverless-performance-comparison-workers-lambda/?hn)

155 points | a17anxx | 1 year ago | 64 comments

Comparing Serverless Performance for CPU Bound Tasks(https://blog.cloudflare.com/serverless-performance-with-cpu-bound-tasks/?h)

127 points | bosdev | 1 year ago | 62 comments

AWS API Performance Comparison: Serverless vs. Containers(https://www.alexdebrie.com/posts/aws-api-performance-comparison/)

123 points | abd12 | 10 months ago | 54 comments

Distribute messages effectively in serverless applications(https://medium.com/epsagon/the-right-way-to-distribute-messages-effectively-in-serverless-applications-f427e4229e67)

11 points | nshap | 2 years ago | 3 comments

In-depth performance analysis of message distribution in serverless and Lambda applications.

“The right way to distribute messages effectively in serverless applications” https://medium.com/epsagon/the-right-way-to-distribute-messages-effectively-in-serverless-applications-f427e4229e67

Serverless Performance: Cloudflare Workers, Lambda and LambdaEdge(https://blog.cloudflare.com/serverless-performance-comparison-workers-lambda/)

7 points | yvoschaap2 | 1 year ago | 0 comments

Serverless Performance: Cloudflare Workers, Lambda and LambdaEdge(https://blog.cloudflare.com/serverless-performance-comparison-workers-lambda/)

5 points | beefman | 1 year ago | 0 comments

Ask HN: Which one to choose: Serverless or Containers?

4 points | sitajay | 2 years ago | 2 comments

I've read about people switching over from Docker (container) to AWS Lambda (serverless) but what scratches my head is, are there any performance (latency and cold start especially) difference? I mean, both are quite similar, so there shouldn't be any difference at all?

Let's say you create a single function that returns a certain calculated number, and you deploy this as Lambda function and to docker. My guess is that both should perform that function equally fast, am I wrong?

My question: Why should you prefer one over the other?

My understanding is that AWS Lambda functions are just like Docker except they're "smaller" and stateless and only runs when the function is being called.

Serverless Performance and Coldstarts(https://thenewstack.io/how-cold-starts-impact-serverless-performance/)

4 points | farrahpdx | 1 year ago | 0 comments

Serverless Artillery: Instant, cheap, and easy performance testing at scale(https://github.com/Nordstrom/serverless-artillery)

2 points | vector_spaces | 9 months ago | 0 comments

AWS API Performance Comparison: Serverless vs. Containers vs. APIG Service Proxy(https://www.alexdebrie.com/posts/aws-api-performance-comparison/)

2 points | abd12 | 10 months ago | 0 comments

Comparing Serverless Performance for CPU Bound Tasks(https://blog.cloudflare.com/serverless-performance-with-cpu-bound-tasks/)

2 points | chx | 1 year ago | 0 comments

Distribute messages effectively in serverless applications

2 points | nshap | 2 years ago | 0 comments

In-depth performance analysis of message distribution in serverless and Lambda applications.

https://medium.com/epsagon/the-right-way-to-distribute-messages-effectively-in-serverless-applications-f427e4229e67

Nuclio: High-Performance Serverless event and data processing platform(https://github.com/nuclio/nuclio)

2 points | qubit23 | 2 years ago | 0 comments

Ask HN: How do you measure your serverless performance?

2 points | sitajay | 2 years ago | 0 comments

Portfolio Performance and Risk Analytics API – Basic

2 points | ruiheh | 2 years ago | 0 comments

Portfolio Performance & Risk Analytics API is a collection of functions for portfolio performance and risk analysis. Built on a serverless architecture and a microservice framework, this package aims to improve data warehouse ETL / ELT speed/accuracy and firms' real-time reporting capabilities.

https://market.mashape.com/AIDVP/portfolio-performance-and-risk-analytics-basic

Canary deployment for serverless

1 points | rsh210m | 2 years ago | 2 comments

All, I'm looking for feedback about a new product me and my team are now building, focusing on testing and deployment of applications involving serverless code. In an effort to increase the speed and predictability of the dev process, we want to enable canary deployment of new code as well as in production testing for serverless functions. Using this method, we want to enable dev teams to efficiently monitor the quality and readiness of their code as an integrated part of their production environment.

To achieve this we implemented an instrumentation of **serverless** functions using an intelligent **serverless** redirection module. This module uses a predefined policy to route events and triggers to achieve a well balanced gradual deployment of new code as well as in-production testing of the code.

As an example, the redirection module may apply a rule where only 5% of the production traffic will be routed through the a newly deployed code from ‘Branch XXX’. Another rule can determine that events that are originating from a defined source (filename/location, user pool, address…) are labeled as “Test-Traffic-Branch-YYY” and routed through a newly deployed code from “Branch YYY”

A dashboard will present all the configured routes (both testing and production) and related statistics for quality, **performance** latency and more.

The idea is to allow a well managed execution of multiple tested routes involving multiple dev pipelines on the same system concurrently, all within (optionally) the actual production context.

Using this framework we also offer a way for developers to work on their code In a controlled lab environment within the production context. In this lab environment, we offer a rich feature-set to enable developers to interact with their code using their own locally installed IDE and other dev tools for monitoring, troubleshooting and rapid deployment of new code changes as may be needed.

Any feedback will be most welcome!

Serverless **Performance**: Cloudflare Workers, Lambda and LambdaEdge(https://blog.cloudflare.com/**serverless**-**performance**-comparison-workers-lambda/)

1 points | alex_young | 1 year ago | 1 comments

Serverless and PHP: **Performance**(http://mnapoli.fr/**serverless**-php-performances/)

1 points | velmu | 2 years ago | 0 comments

Serverlessconf New York: Monitoring **Serverless** **Performance** to Manage Cost(https://thenewstack.io/serverlessconf-new-york-monitoring-**serverless**-**performance**-manage-cost/?utm_content=buffer83b40&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer)

1 points | rbanffy | 2 years ago | 0 comments

Nuclio: High-**Performance** **Serverless** event and data processing framework(https://github.com/nuclio/nuclio)

1 points | blopeur | 2 years ago | 0 comments

Benchmarking **Serverless**: IBM Devises a Test Suite to Quantify **Performance**(https://thenewstack.io/ibm-scientists-set-quantify-**serverless**-**performance**/?utm_content=bufferc7a60&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer)

1 points | rbanffy | 2 years ago | 0 comments

Ask HN: Actor model vs. **serverless**/function architectures?

1 points | hurricaneSlider | 3 years ago | 0 comments

For a recent project we developed our application atop Akka.Net (a C# implementation of an actor system) We had a great experience with it despite the architectural learning curve.

Serverless or function apps seem to be quite popular at the moment. But in my mind they appear to be simply a specialised case of the actor model. I've not used **serverless** in production, but at a glance, actor systems seem to win in terms of flexibility, theoretical soundness and maturity of tooling. In contrast I've heard horror stories about the kinds of hacks developers who've embraced function apps have made in order to squeeze out decent **performance**.

A compelling argument in favour of function apps appears to be: cost, hosted services providing an easy means to scale out and relatedly, platform buy in from AWS, Azure, et al. But equally I could easily imagine a PaaS version of an actor system which could share the same attributes.

Is it the language that actor systems are couched in? What makes "**serverless**" more compelling?

Ask HN: Should we discontinue our Android and iOS SDK?

5 points | nawazdhandala | 3 years ago | 2 comments

I'm Nawaz and I work at CloudBoost.io. We're a **Serverless** + Backend as a Service company. Most of our customers use our JavaScript SDK to build React, ReactNative, Angular, NodeJS, Ionic apps and we see less and less traction on our Android and iOS side of things since JavaScript apps can run on mobile too (with Ionic and React Native). Do you think that traction of Java / Swift on phones will decline as JavaScript gains native **performance** on phones.

It takes quite a lot of investment to maintain those side of things and we're at a point where investing in Java / Swift SDK's is not viable.

Let me know what you think?

HTML5-ish window.remoteStorage with openkeyval.js

3 points | markchristian | 9 years ago | 1 comments

Howdy, Hackers; Last week, dustball and I announced OpenKeyval.org here (http://news.ycombinator.com/item?id=1846088), and we got some great feedback. One thing we heard from a few people was a desire for it to be easier to work with from JavaScript. That was something I was very happy to hear, since this idea came out of my own brainstorming around making **server-less** web apps that could still have non-local state.

With that in mind, I'd like to introduce openkeyval.js, which offers an interface to OpenKeyval.org that's vaguely modeled after HTML5's local storage.

// Here's how you set a value window.remoteStorage.setItem('location', 'the couch', optionalCallbackFunction);

// Here's how you retrieve a value window.remoteStorage.getItem('location', function(value, key) { alert('The value of ' + key ' + is ' + value); });

Take a look at the site for more details and a link to the code. We're hosting a minified version on a CDN (incidentally, http://maxcdn.com is pretty awesome), so you can feel free to link directly to it from your sites with a reasonable expectation of reliability and speed.

As always, thanks for reading and please let us know what you think.

PS: This is definitely a first revision. Although it has memoization, it doesn't have any provision for local caching, and it can only get/set one item at a time -- basically, there is definitely a lot of room for **performance** improvements.

Cloudflare Workers is 4.6x faster for CPU-bound tasks than Amazon Lambda(https://blog.cloudflare.com/**serverless**-**performance**-with-cpu-bound-tasks/?hn)

30 points | zackbloom | 1 year ago | 5 comments

AWS Lambda **Performance** Optimization and Monitoring with Tracing and Spans(https://**serverless**.com/blog/aws-lambda-**performance**-optimization/)

1 points | worldsoup | 1 month ago | 0 comments

An interview on the **performance** challenges of FaaS cloud architectures(https://medium.com/jexia/why-**serverless**-is-still-in-its-infancy-311e932 a33cd)

15 points | Jexia | 8 months ago | 0 comments

Performance boost and cost savings for DynamoDB(https://cloudonaut.io/serverless-cache-for-dynamodb-with-elasticache/)

3 points | hellomichibye | 3 years ago | 0 comments

Show HN: ProcessMaker.IO is a workflow engine in the cloud. Need candid feedback

16 points | tdondich | 2 years ago | 7 comments

Our project is at: https://www.processmaker.io/

We built a high performance workflow engine in the cloud aimed for developers. It originally was intended as a BPMN focused engine but you can also create complex workflows programatically without any BPMN diagramming tools. Think of it as a platform that goes well beyond ITTT or Zapier. This platform means we can do services orchestration and serverless programming (using our script tasks which currently support LUA) as well as human enabled workflows (think of using forms and approvals, etc). We have SDKs in most major languages and we're aiming specifically at developers as the target audience. This would be perfect for ISVs looking to embed complex workflows in their product but not have to build the workflow features themselves.

The problem I'm seeing is that understanding workflow from the beginning is a hard concept. The on-boarding process for a developer to understand the value and to immediately get started is something we're trying to get over. We have various use case examples and video demos that show how you could use it, but I still don't feel comfortable yet with the 1st time experience.

So I'm sharing in hopes that anyone can provide feedback/thoughts? Or even if you have questions on how things are done (which would help us with our messaging). Docs are updated and you can sign up and fire up a workflow engine right away to play with (it's free right now so we can get people playing with it). It's at www.processmaker.io and any suggestions/advice for improving the messaging on the website is also really appreciated.

The project is in public Alpha right now, so if something isn't working right, please bring it up so I can have the team fix it up!

Thanks all.

Ask HN: What tech stack would you choose to bootstrap a side project in 2020?

10 points | yagodragon | 3 days ago | 11 comments

I'm a college student trying to choose a language/framework to build some side projects. Other than the classic CS languages(java,python,c/c++), I've learned some JS and Vue.js and now I'm looking into a proper backend language and framework.

After searching online and asking friends I've gathered some feedback on the most popular solutions.

Rails: Proven and mature. 3rd party gems that can help you with anything web-related. The ecosystem is thriving and most developers are happy using it.

Laravel: Made PHP cool again. Took lessons from Rails and probably provides the best Developer Experience when it comes to building a monolith app from development to deployment. PHP is also the most popular backend language.

Django: People love python. I hear that Django Rest Framework is a great tool, plus, the ability to add ml features on your existing app is a big plus.

Node.js: Full-stack Javascript is great but the backend landscape is a mess. Probably a thin backend without complex rails-like structure.

Phoenix/Go/Rust: Good options for specific use cases where performance matters.

Java/C#: Complex and enterprisey. C# is gaining traction but the 3rd party ecosystem is still lacking behind other options.

Backendless: SPAs, Next.js/Nuxt.js/Ember, JAMstack. Use services like Firebase/Auth0 and 3rd party APIs for backend logic or serverless functions.

What would you suggest someone learn in 2020? I know the answer is "it depends" but I know that for example Rails, Laravel and Django are basically tools for the same job, building CRUD applications on the server. Would you suggest someone to learn Ruby or PHP in 2020, though many consider them dead?

- 1
- 2

About • Setting • Help • API Documentation • Hacker News • Fork/Contribute • Cool Apps