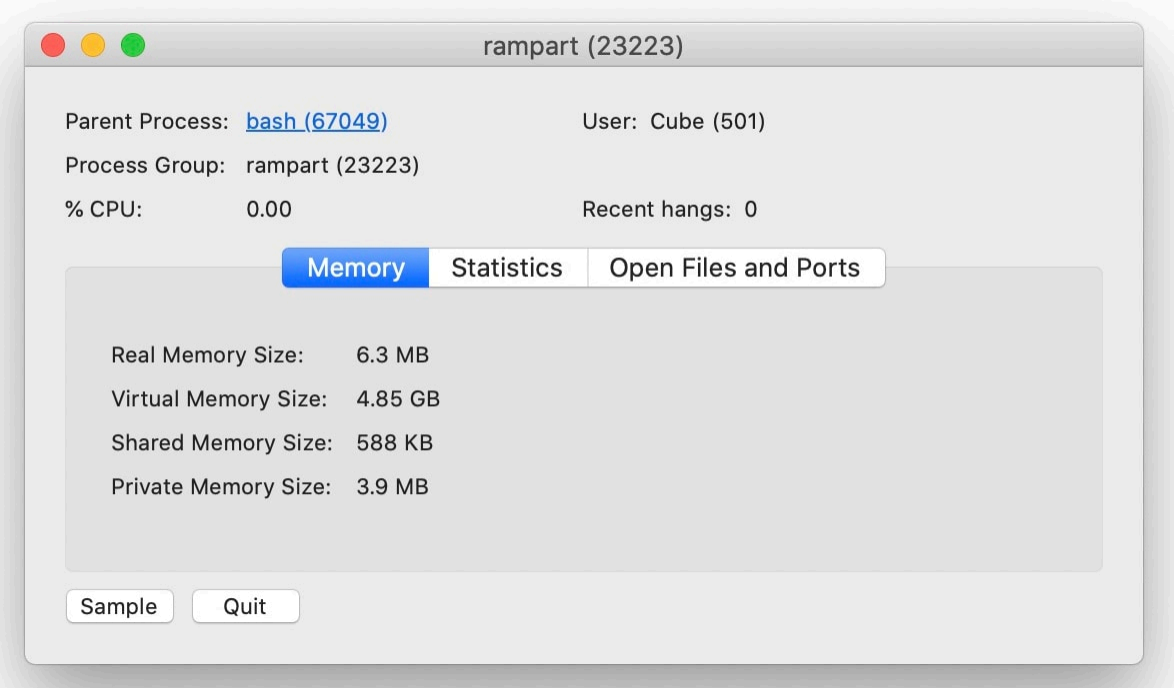
Squish the Stack

Rampart is a javascript based complete stack development environment which is extremely resource stingy. This entire site is running on a Raspberry Pi Zero (version 1). Rampart is also fast and portable; it will run on nearly any *nix OS and benchmarks nearly as fast as Nginx on static content. It's also free to use, modify, and redistribute in most use cases.

We're not kidding about it being resource stingy. Here's the Rampart HTTP server running on my machine as I'm coding this web page on an Intel Mac with almost all of its modules loaded:



Full Text Search with SQL

We've licensed the same fulltext and relational database engine that

HTTP, HTTPS, and Websockets

Rampart's HTTP/HTTPS server is based on a modded libevhtp_ws and

good at serving database driven content. It is also realtime so that new or changed documents are instantly searchable. We've included a CSV parser so you can migrate data into it as well.

more info...

CURL Module

Rampart provides a robust interface to the Curl library, giving you the ability to fetch URLs, post forms, or send email and attachments easily. It includes all the functionality you've come to expect with the Curl command line utility, but from within Rampart JavaScript. The combination of this module and the HTML Parser provide a fairly robust data scraping tool set.

more info...

REDIS Client

We've included direct REDIS integration for all the obvious reasons one might want to use it. We also crafted a custom XREAD command to work like PUB/SUB so you can, in combination with WebSockets, create chats, channels and DMs that don't require the extra step of saving, retrieving and sorting messages from a message queue.

more info...

chose it. We haven't benchmarked it against Node yet but it's reasonable to assume we'll keep pace while consuming a lot less of the host's resources

more info...

Threading

Rampart threads let you run functions not only asynchronously but in also in a multi-threaded environment. Running a function in a thread is as easy as running one in a **`setTimeout()`**. Each thread is run in its own isolated JavaScript interpreter and variables are easily shared at any time via a clipboard. Running an asynchronous callback in a thread when a variable is copied to the clipboard is also possible.

more info...

Lightning Memory-mapped Database

LMDB is a fast, truly ACID compliant RAM based key-value store. It's read biased and serves binary or text data really well. We've added automatic conversion to/from CBOR and JSON to make it that much easier. It's not as fast as Redis when in full ACID mode, but then again it'll never lose data when Redis might. LMDB startup is relatively instant, even on large

Markdown Engine

The markdown engine uses the Cmark library to implement the CommonMark syntax. This allows you to efficiently convert to HTML on the server side and have symmetry with the CommonMark client-side javascript module. This is handy for things like blog entries, documentation, and CMS page markup. *Note: We intend to eventually build and use our own CMS using it.*

more info...

ROBOTS.TXT

Rampart gives you the power to crawl and scrape in a very flexible manner. However, with great web crawlers come great responsibilities. The ROBOTS.TXT module uses Google's robots.txt code so you can avoid abusing sites that do not wish to have portions of their content crawled.

more info...

HTML Parser

Using a modded version of HTMLTidy Rampart provides the ability to manipulate and traverse HTML in a manner somewhat akin to JQuery, but on the server side. This module can also clean up and format messy or compacted HTML after it's been CURL'ed from somewhere. *Note: I used this module several times to locate errant imbalaced s when writing this site.*

more info...

Cryptography

The OpenSSL library module provides the ability to safely encrypt and decrypt stuff generate crypto-hashes, generate RSA key pairs, and more. It also provides cryptographic quality random and Gaussian random number generation. Sorry, it doesn't generate crypto-currency though. *Note: for speedy non-crypographic hashing see the non-cryptographic hashes.*

more info...

Networking

The rampart-net module provides access to low level, asynchronous TCP/IP calls for creating client and server applications, as well as name resolution. The functions are

Python

Yes we embedded the c-python interpreter into a Rampart module. It allows you to load python modules, run python scripts and manipulate python variables in JavaScript without a

more info...

functionality that Rampart doesn't (yet) provide, this can get your there. See this example for a quick rundown of what is possible.

more info...

OS Utilities

All the common commands and functions required to manipulate files and talk to the OS are in here *(hopefully)*: **`printf()`**, **`sprintf()`**, **`fopen()`**, **`fread()`**, **`fwrite()`**, **`hexify()`**, **`dehexify()`**, **`trim()`**, **`stat()`**, **`exec()`**, **`shell()`**, **`kill()`**, **`mkdir()`**, **`rmdir()`**, **`sleep()`**... and a lot more.

more info...

Fast HyperLogLog, non-crypto Hash, & Random

We implemented our own super-fast HyperLogLog algorithm for on-line counting of distinct things. e.g. unique IP addresses. It's substantially faster and more accurate than Redis's. Our HLL uses Google's City Hash algorithm internally, but we also expose City Hash and its cousin Murmur Hash, as well as a fast pseudo random number generator based on XorShift64.

more info...

More to come

We've been working on assembling Rampart for a beta release since the start of Covid. Some stuff hasn't made it into it yet, but we are continuing to add features and by no means are we nearly done adding things. There are also tutorials and unsupported extras which can give you an idea of the direction we are heading.

Why did we do this?

machine. Additionally, it's not that much fun.

In creating Rampart we sought to create a "full stack" that was simple and fast to download, configure, and start creating whatever application you had in mind in a time frame measured in minutes rather than days. User machines, phones, and IOT devices need to serve their primary role without impediment. Rampart's job is to support that role and not get in the way. OTOH if you dedicate a machine or cluster to a Rampart task you'll get a lot more done with a lot less investment.

We chose the DukTape javascript engine over the V8 engine because while V8 is very fast, it's also kind of a RAM and CPU pig. Our philosophy was to do everything difficult or complex in C and let javascript be the fun-glue. That way it really didn't matter what engine was chosen. Almost every module is coded in C as is the DukTape interpreter. This makes everything extremely portable with a minimum amount of fuss. We include the Babel transpiler to drag DukTape into ECMAScript 2015+ land. Rampart checks at runtime if you've changed anything and then transpiles automatically behind the scenes.

Who are we?

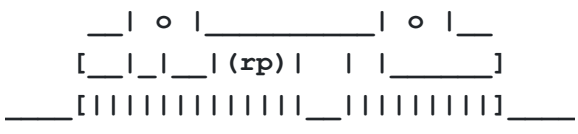
Moat Crossing Systems LLC, the makers of Rampart, is a couple of guys who between them have developed software and applications that have served many billions of pages of database backed content on the internet. We had some summer help from a couple of Columbia students who helped up keep tabs on the "current-think" and coded some of the OS Utility functions. It all started in January of 2020 with a phone conversation where we were bitching about how un-fun web development had become, and how resource heavy everything seemed to be. A comment was made that if we could full-text index Wikipedia and serve it up on a Raspberry Pi Zero, then we'd have something unique that people might want. So that's what we did.

We hope you will give this thing a try. Contact us here or at `support@rampart.dev` if you have questions or need help with something.

The Raspberry Bush

Here's a photo of our fancy server farm before it was installed in a rack at the ISP. There's four Pi Zeros and two Pi 4s. The four Zeros are mirror copies of one another just in case *stuff* happens. The two Pi 4's allow us to do development and builds a bit faster than the Zeros allow us. The "bush" is proxied by an NginX server running on a fairly large old Xeon machine. This made certificate managment and other things somewhat easier on us.

Please note that we have no particular bias towards producing software principally for the Raspberry Pi. The whole purpose for its usage here is to demostrate how much Rampart can do with minimimal resources. That way you can extrapolate how fast and efficient it will be on larger Intel, AMD, and ARM platforms.



RAMPART

View JS Source