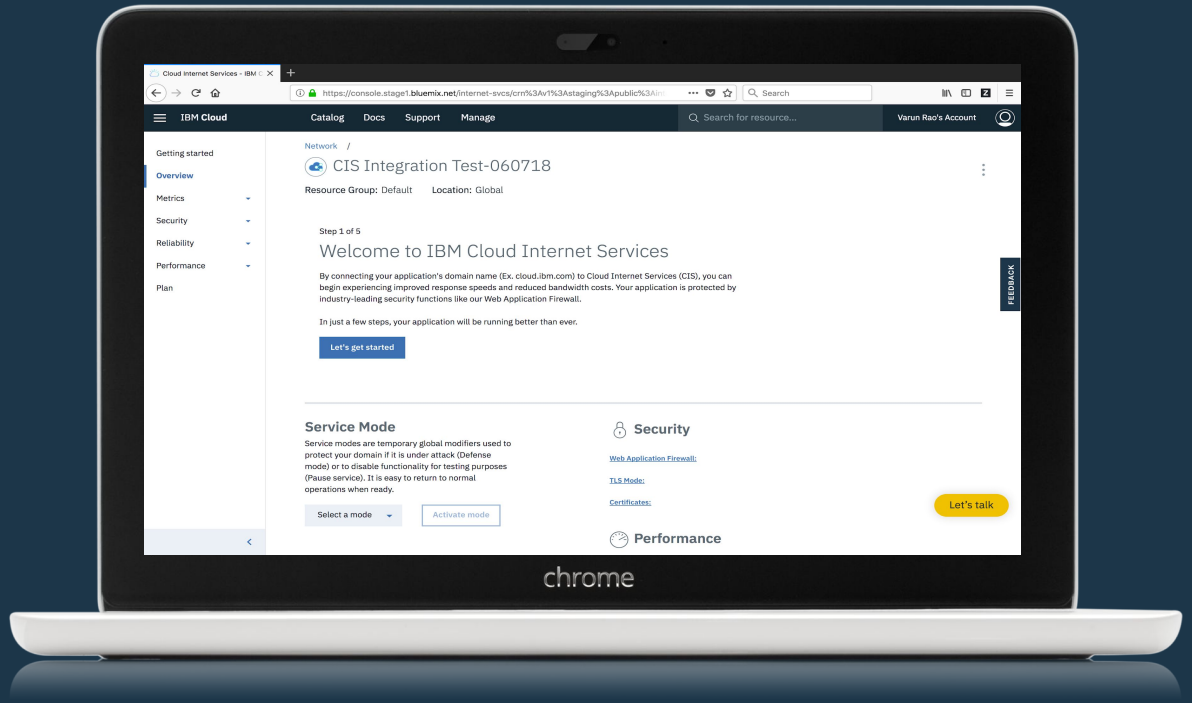
Performance improvements for Cloud Internet Services

Varun Rao & Disaiah Bennett

Manager: Nikhil Gupta - Team: 7 - Austin, Texas

Cloud Internet Services

Our team's API delivered security, reliability, and high performance services from Cloudflare to IBM's clients using IBM cloud.



What problem were we solving?

- We use Kubernetes and Armada to deploy our service.
- We needed to deliver our services at scale, and our API needed to be able to handle hundreds of requests per second.
- Our old solution was slow, so we had to improve the performance of our team's service.

So what did we do?

– – –

- The biggest bottleneck in any internet service will always be communication between servers.
- We designed and developed a Python module for making HTTP requests with libcurl instead of the most commonly used open source Requests library or the built in Python standard library.

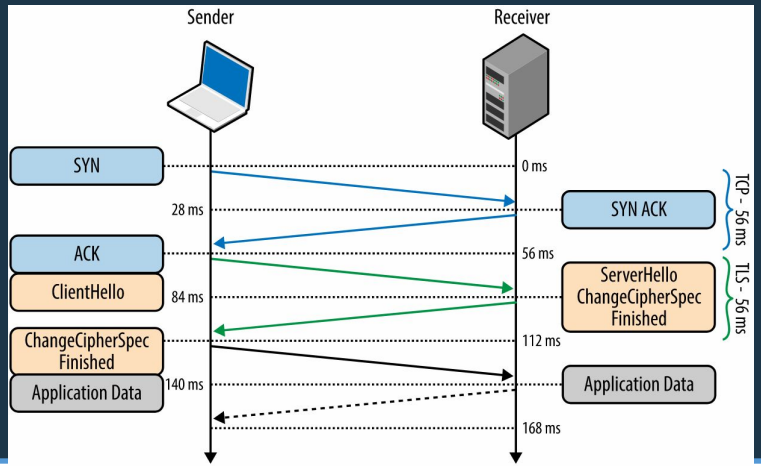
So how much faster is our version?

154%

So why is ours faster?

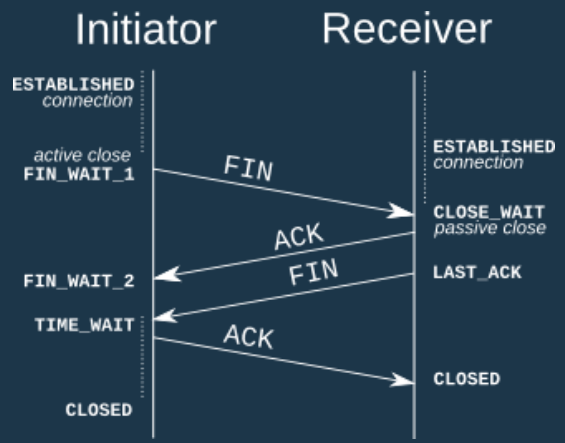
TCP connection reuse

- When you go to any new website or server, a new TCP connection and is opened between your computer and that server, which requires a new TLS handshake to be performed.
- Opening this connection can take a couple seconds, easily longer than the entire data transfer.



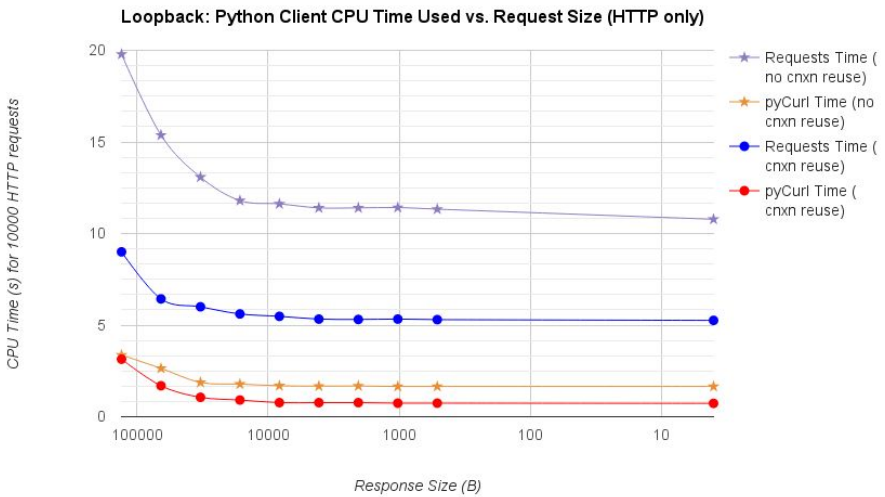
Tracking closed TCP connections

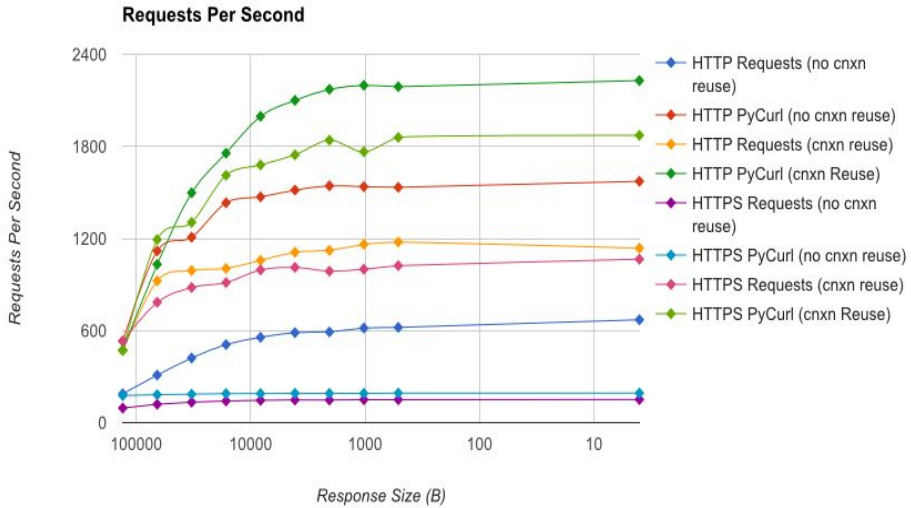
- Our original request solution also didn't track closed TCP connections.
- Every so often the OS closes a TCP connection, and if we keep trying to make requests using it, we will just fail.
- We managed to fix this by using curl to make requests.

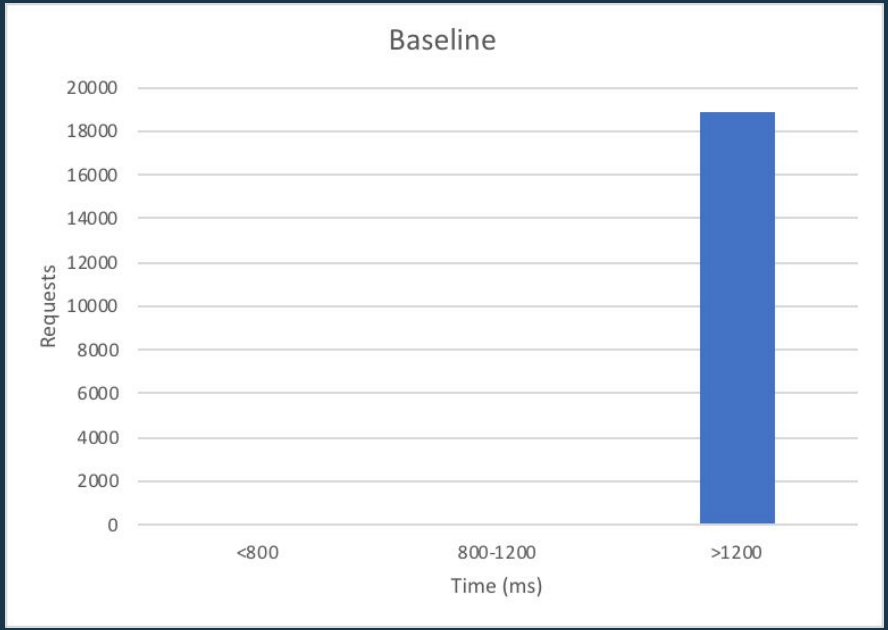


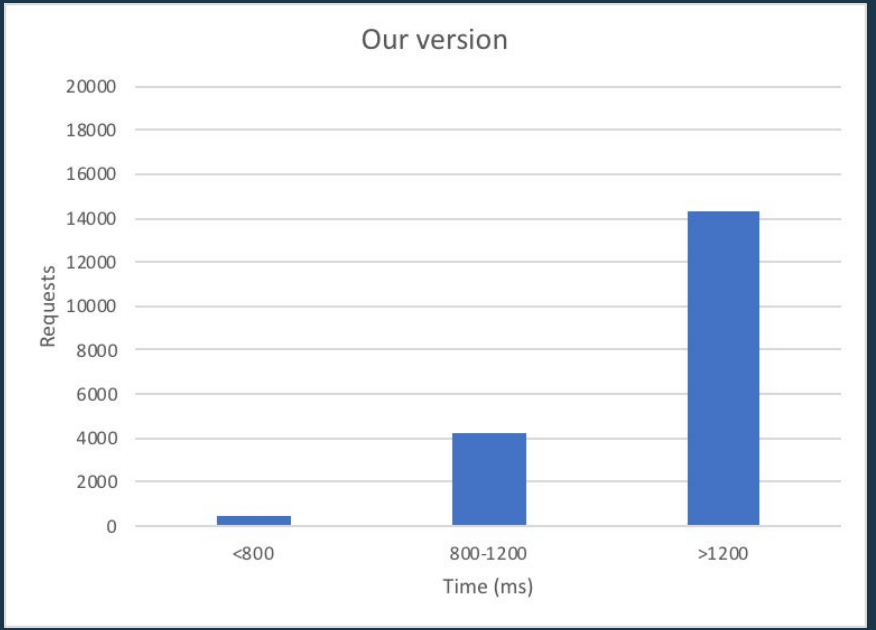
PyCurl vs Requests

- Instead of relying on Python standard libraries, we made HTTP requests using libcurl, which is written in C, which is way faster than Python.









Old version

No connection reuse, no TCP connection tracking, and using Requests library.

New version

Connection reuse, TCP connection tracking, and using libcurl.

Like data?

- Our slowest request were no faster than the old slowest requests. We rely on a lot of external APIs, so this slowdown could be caused by any number of issues.
- Our median request time is more than twice as fast as the old median, but we didn't see twice the improvement due to the slowest requests still being very slow.

Old baseline

Median Baseline

3493

STATISTICS — Expand all groups | Collapse all groups

Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Req/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	19000	19000	0	0%	109.195	375	3493	4941	9035	14522	23363	4289	2599
create zone	500	500	0	0%	2.874	2455	10780	14526	21445	21857	23363	11878	5512
create dns records	5000	5000	0	0%	28.736	1361	3553	5046	11190	14844	17851	4674	2881
list records	1500	1500	0	0%	8.621	1322	3300	4331	6861	8412	10292	3687	1552
change d...de to on	500	500	0	0%	2.874	1424	3824	6527	8270	9852	10719	4692	2092
update r...ds by id	5000	5000	0	0%	28.736	1310	3442	4447	6489	8384	11260	3770	1400
change d...e to off	500	500	0	0%	2.874	1189	3909	5933	7359	8756	10448	4382	1798
purge all caches	500	500	0	0%	2.874	1093	3197	5130	6978	8420	8710	3822	1681
delete r...ds by id	5000	5000	0	0%	28.736	375	3218	4559	7257	9233	12270	3703	1767
delete zone	500	500	0	0%	2.874	1347	5144	7407	10354	11795	14274	5691	2494

New Median

1587

Our version

STATISTICS — Expand all groups | Collapse all groups

Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Req/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	19000	19000	0	0%	168.142	212	1587	2063	5725	24686	48450	2546	3956
create zone	500	500	0	0%	4.425	3137	18129	26844	38267	45172	48450	20228	10510
create dns records	5000	5000	0	0%	44.248	737	1974	2470	8937	19260	27709	3033	3456
list records	1500	1500	0	0%	13.274	665	1458	1746	4806	5437	6081	1883	1178
change d...de to on	500	500	0	0%	4.425	945	1746	2159	5826	6350	6628	2398	1577
update r...ds by id	5000	5000	0	0%	44.248	749	1753	2031	5115	5595	6259	2092	1105
change d...e to off	500	500	0	0%	4.425	862	1434	1562	3527	4900	5332	1575	736
purge all caches	500	500	0	0%	4.425	414	1161	1271	1563	3439	4949	1227	434
delete r...ds by id	5000	5000	0	0%	44.248	212	1077	1274	3495	4610	5418	1301	841
delete zone	500	500	0	0%	4.425	528	1359	1549	2734	3147	4326	1424	574

Contact

Varun Rao
varunsatishrao@gmail.com
https://github.com/varunsrao

Disaiah Bennett
lavontae.bennett@gmail.com

And thanks to our mentor, Andrei Ta!