
WebFrame++

C++ web application framework



Алекс Цветанов
Софийска Математическа Гимназия, София, България

Съдържание

1	Въведение	4
2	Проблем	5
2.1	При ученици	5
2.2	При състезатели	6
2.3	В бизнеса	7
3	Сравнение	7
3.1	Синтаксис	7
3.1.1	Общ поглед върху интерфейса	7
3.1.2	Flask - Python	8
3.1.3	Express - Node.JS	8
3.1.4	Извод	9
3.2	Ефективност	9
3.2.1	Hello, World!	9
3.2.2	Сума на N произволни числа	13
3.2.3	Извод	16
4	Технологии	18
5	Бъдеща разработка и разпространение	18
6	Благодарности	19

Абстракт

WebFrame++ е съвременна библиотека, която цели да направи един от най-бързите програмни езици (C++) удобен за разработка на интернет приложения.

Разработката предоставя удобен синтаксис за създаване на уеб сървър, задаване на пътища и тяхното съдържание по начин, наподобяващ този на повечето използвани в момента библиотеки и модули със същата цел, но за други езици.

Примери за такива са Express.JS за Node.JS, Flask и Django за Python, дори Spring framework за Java.

WebFrame++ is up-to-the-minute library that aims to make one of the fastest programming languages (C++) efficient for web applications.

The library provides a convenient syntax for creating a web server, setting up paths and content in a way similar the most up-to-date libraries and modules with the same purpose, but in other languages.

Examples of these are Express.JS for Node.JS, Flask and Django for Python and even Spring framework for Java.

1 Въведение

Проблемът на всеки разработчик на интернет приложения на C++ е, че трябва да използва стари и бавни методи като CGI и FastCGI, които са значително по-неефективни от наложените в момента технологии в тази област. Разработчиците имат и опцията да използват socket (това е метод от езика C, който работи на значително по-ниско ниво) и не използва модернизациите в съвременните версии на C++ - C++20, C++17, дори C++14 и C++11, което го прави труден за използване от програмистите.

От друга страна езикът C++ е широко разпространен в сфери като търсачки, сложни алгоритми, изкуствен интелект, големи данни и други. Това означава, че благодарение на WebFrame++ разработчиците, не само ще могат да съчетават всички тези тематики с уеб приложения, но и с REST APIs за мобилните си приложения, които също може да бъдат имплементирани на C++.

WebFrame++ е проект, разработен в полза, както на големи и средни, така и на малки компании, които се занимават с разработката на интернет приложения. Библиотеката е подходяща за употреба в училища и университети в България, тъй като там се изучава програмният език C++, което ще мотивира още повече хора да се занимават професионално с езика. Също така библиотеката подпомага разрастването на C++ обществото по целия свят.

От гледна точка на бизнеса библиотеката ще намали очакваното ниво на знания за работниците, тъй като няма да има нужда те да могат да четат/разбират и пишат на езици, различни от C++ (за сървърната част), стандартните HTML, CSS и JavaScript (за изгледа на страниците), бази данни (най-често срещаните са SQL и MongoDB) и евентуално JSON, ако се налага в съответната компания.

Наличието на повече кандидати с достатъчни знания, съответно, води до по-бързо разрастване на компанията и по-голяма конкурентоспособност, както вътрешно между програмистите във фирмата, така и на компанията в сравнение с другите.

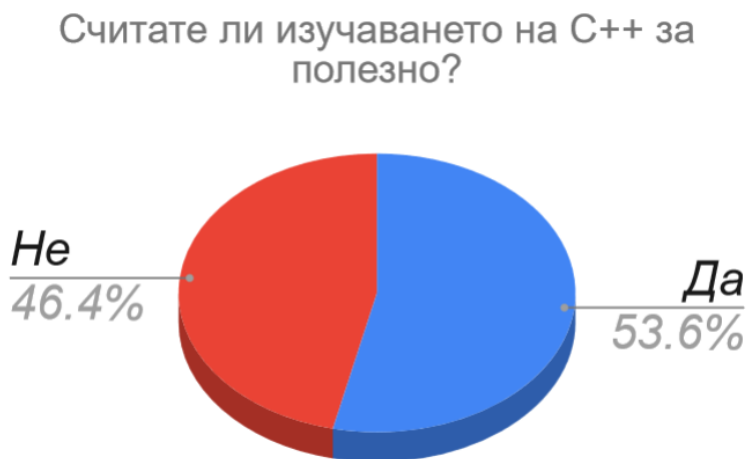
2 Проблем

Ще разгледаме проблемите, които решава библиотеката от три гледни точки:

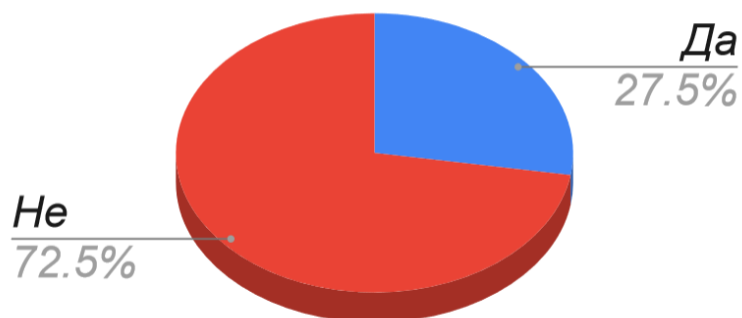
2.1 При ученици

В повечето профилирани училища, учениците се сблъскват с програмния език C++, наложил се като лесноусвоим от тийнейджърите.

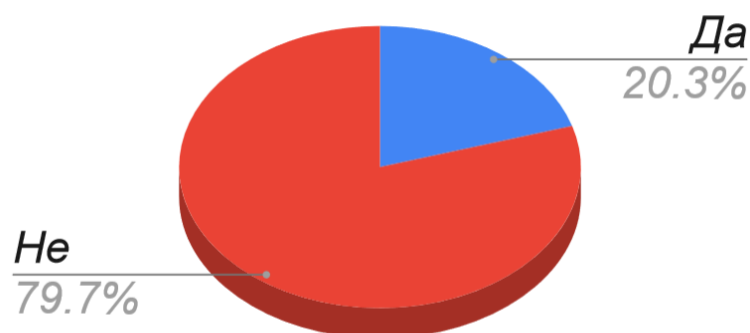
На фона на всичко позитивно учениците не виждат приложната страна на езика, който се е наложил като един от най-ефективните (ако не и най-ефективният) за разработка на игри. Проведох непредставително проучване сред ученици от 10. и 11. клас на Софийската математическа гимназия.[9] Ето и резултатите от него:



Представяте ли си да създадете
собствена игра на C++?



Представяте ли си да създадете собствен
сайт на C++?



Създаването на сайт на C++ със знания, придобити по време на учебния процес, значително повишава интереса към развитието в тази област и изучаването на този език.

2.2 При състезатели

По националните състезания по информатика се забелязва тенденция на демотивация на участниците с тенденциозно място извън челните. Те вече знаят програмният език достатъчно добре и библиотеката ще им позволи бързо да пренасочат знанията си към приложната сфера, където състезателите могат да са на челните места, впечатлявайки със скоростта на продукта си. (3) Освен това времето, което ще е необходимо за „преориентиране“ към библиотеката, е минимално.

2.3 В бизнеса

Благодарение на интерфейса на библиотеката, бизнесът може много бързо да мигрира към нея. Използването ѝ ще допринесе за разрастването на компаниите, поради наличието на ученици и студенти, които ще завършат със знания за технологията, което ще доведе до повече квалифицирани кадри.

WebFrame++ се оказва по-успешната технология за разработка на уеб приложения от гледна точка на скоростта на платформата. Повече информация може да намерите в точка „3. Сравнение“.

3 Сравнение

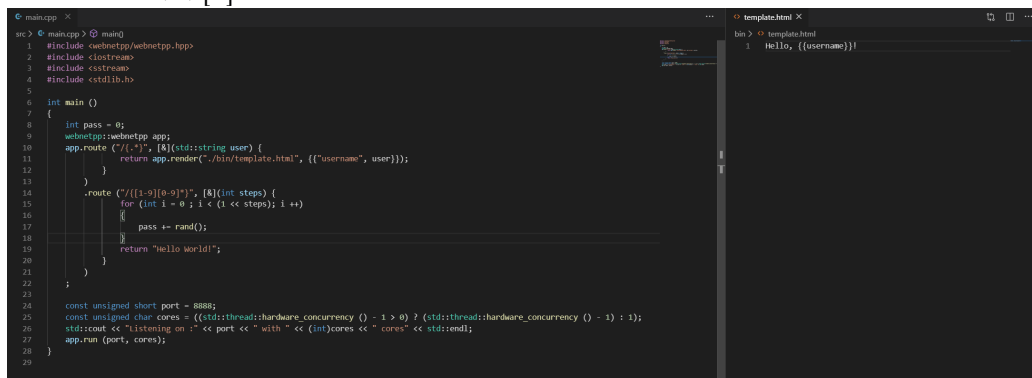
WebFrame++ е библиотека за програмния език C++, която се стреми да наподобява синтаксиса на най-използваните съвременни технологии за разработка на уеб приложения. Разбира се, имайки предвид това, че C++ е един от езиците, работещи на „ниско ниво“, се целим и в ефективност на библиотеката.

Ефективността на технологиите за разработка на уеб приложения се базира основно на времето, необходимо за реакция на дадена заявка от клиента.

3.1 Синтаксис

3.1.1 Общ поглед върху интерфейса

По-долу е показан фрагмент от код, показващ начина на употреба на WebFrame++[8].



```
1 #include <webnetpp/webnetpp.hpp>
2 #include <iostream>
3 #include <sstream>
4 #include <stdlib.h>
5
6 int main ()
7 {
8     int pass = 0;
9     webnetpp::webnetpp app;
10    app.route("/", [&](std::string user) {
11        return app.render("../bin/template.html", {{"username", user}});
12    })
13    .route("/[0-9]{0-9}*", [&](int steps) {
14        for (int i = 0; i < (1 << steps); i++)
15            pass += rand();
16        return "hello world!";
17    })
18    .route("/hello world!");
19
20    const unsigned short port = 8080;
21    const unsigned char cores = ((std::thread::hardware_concurrency() - 1 > 0) ? (std::thread::hardware_concurrency() - 1) : 1);
22    std::cout << "listening on " << port << " with " << (int)cores << " cores" << std::endl;
23    app.run(port, cores);
24}
```

```
1 Hello, {{username}}!
```

3.1.2 Flask - Python

По-долу е показан фрагмент от код, показващ начина на употреба на конкурента Flask Framework за Python.

```
1 | from flask import Flask
2 | import random
3 | app = Flask(__name__)
4 |
5 | @app.route("/<steps>")
6 | def hello(steps):
7 |     x = 0
8 |     for i in range(2 ** int(steps)):
9 |         x += random.randint(0, 1e9)
10 |        x %= 1e9
11 |        return "Hello World!"
12 |
13 | if __name__ == "__main__":
14 |     app.run()
15 |
```

3.1.3 Express - Node.JS

По-долу е показан фрагмент от код, показващ начина на употреба на конкурента Express за Node.JS.

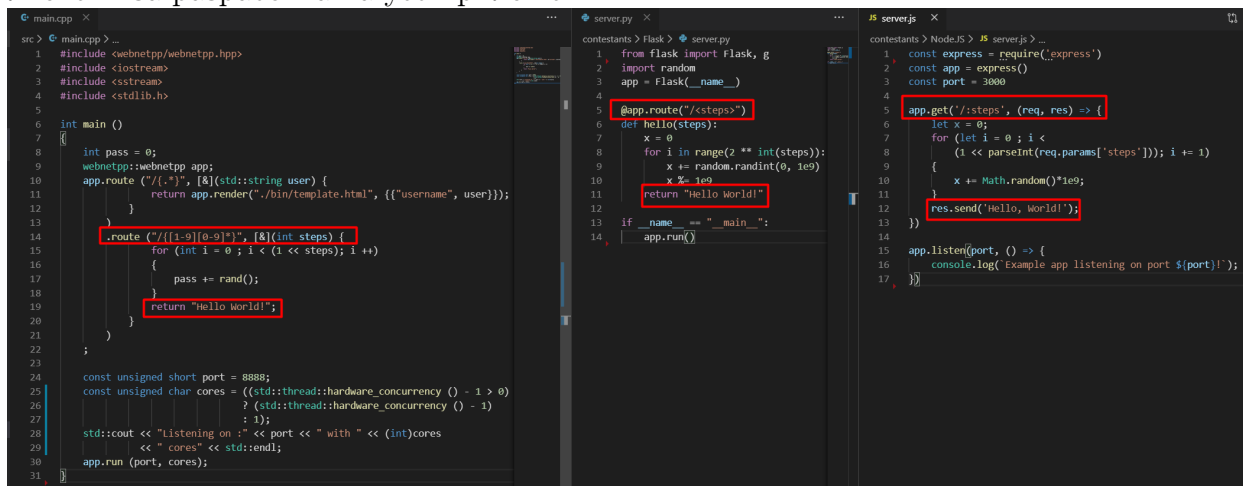
```
const express = require('express')
const app = express()
const port = 3000

app.get('/:steps', (req, res) => {
  let x = 0;
  for (let i = 0 ; i < (1 << parseInt(req.params['steps'])); i += 1)
  {
    x += Math.random()*1e9;
  }
  res.send('Hello, World!');
})

app.listen(port, () => console.log(`Example app listening on port ${port}!`))
```


3.1.4 Извод

Библиотеката наподобява най-разпространените в световен мащаб библиотеки за разработка на уеб приложения.



Библиотеката тепърва ще се опростява все повече и повече. В момента съдържа голяма част от възможностите на лидерите в областта и като предимство има допълнителни опции, които улесняват програмиста и му спестяват доста проверки за валидност на данните.

3.2 Ефективност

С цел сравнение на ефективността са стартирани 3 сървъра (на `WebFrame++`, на `Flask` и на `Express`) и изпращаме еднакви заявки към тях, като измерваме и времето за реакция на всеки сървър. Резултатите са:

3.2.1 Hello, World!

Първият тест, който бе направен се състои в това да се „изциклят“ числата от 1 до n и след това да се върне като резултат от заявката символния низ „Hello, World!“. С псевдо код този тест може да се представи като изпълнение на фрагмента:

Algorithm 1

```
1: procedure Request
2:   n ← number
3: begin:
4:   i := 0
5: loop:
6:   if i == n then return "Hello, World!"
7:   if i < n then
8:     i := i + 1
9:     goto loop.
```

3.2.1.1 Сравнение с Flask (Python)

Таблица 1: Сравнение м/у WebFrame++ и Flask

Стъпки на цикъл преди връщане на "Hello, World!"	WebFrame++	Flask	Сравнение м/у WebFrame++ и Flask	
2	0m0.235s	0m0.226s	-0.0002m	-3.83%
4	0m0.227s	0m0.227s	0.0000m	0.00%
8	0m0.224s	0m0.230s	0.0001m	2.68%
16	0m0.228s	0m0.227s	0.0000m	-0.44%
32	0m0.226s	0m0.236s	0.0002m	4.42%
64	0m0.229s	0m0.228s	0.0000m	-0.44%
128	0m0.226s	0m0.226s	0.0000m	0.00%
256	0m0.228s	0m0.227s	0.0000m	-0.44%
512	0m0.226s	0m0.226s	0.0000m	0.00%
1024	0m0.226s	0m0.227s	0.0000m	0.44%
2048	0m0.228s	0m0.228s	0.0000m	0.00%
4096	0m0.228s	0m0.226s	0.0000m	-0.88%
8192	0m0.225s	0m0.226s	0.0000m	0.44%
16384	0m0.225s	0m0.226s	0.0000m	0.44%
32768	0m0.225s	0m0.226s	0.0000m	0.44%
65536	0m0.225s	0m0.227s	0.0000m	0.89%
131072	0m0.226s	0m0.233s	0.0001m	3.10%
262144	0m0.227s	0m0.238s	0.0002m	4.85%
524288	0m0.228s	0m0.250s	0.0004m	9.65%
Продължава на следващата страница				

Таблица 1 – продължение от предишната страница

Стъпки на цикъл преди връщане на "Hello, World!"	WebFrame++	Flask	Сравнение м/у WebFrame++ и Flask	
1048576	0m0.229s	0m0.277s	0.0008m	20.96%
2097152	0m0.231s	0m0.290s	0.0010m	25.54%
4194304	0m0.239s	0m0.359s	0.0020m	50.21%
8388608	0m0.242s	0m0.458s	0.0036m	89.26%
16777216	0m0.258s	0m0.692s	0.0072m	168.22%
33554432	0m0.290s	0m1.237s	0.0158m	326.55%
67108864	0m0.360s	0m2.163s	0.0301m	500.83%
134217728	0m0.475s	0m4.102s	0.0605m	763.58%
268435456	0m0.716s	0m7.896s	0.1197m	1002.79%
536870912	0m1.217s	0m15.572s	0.2393m	1179.54%
1073741824	0m2.199s	0m31.241s	0.4840m	1320.69%
Най-голямото изоставане на WebFrame++			-0.0002m	-3.83%
Най-големият аванс на WebFrame++			0.4840m	1320.69%
Най-малкото изоставане на WebFrame++			0.0000m	-0.44%
Най-малкия аванс на WebFrame++			0.0000m	0.44%

3.2.1.2 Сравнение с Express (Node.JS)

Таблица 2: Сравнение м/у WebFrame++ и Express

Стъпки на цикъл преди връщане на "Hello, World!"	WebFrame++	Node.JS Express	Сравнение м/у WebFrame++ и Express	
2	0m0.235s	0m0.033s	-0.0034m	-85.96%
4	0m0.227s	0m0.028s	-0.0033m	-87.67%
8	0m0.224s	0m0.027s	-0.0033m	-87.95%
16	0m0.228s	0m0.030s	-0.0033m	-86.84%
32	0m0.226s	0m0.030s	-0.0033m	-86.73%
64	0m0.229s	0m0.026s	-0.0034m	-88.65%
128	0m0.226s	0m0.026s	-0.0033m	-88.50%
256	0m0.228s	0m0.031s	-0.0033m	-86.40%
512	0m0.226s	0m0.026s	-0.0033m	-88.50%
1024	0m0.226s	0m0.030s	-0.0033m	-86.73%
Продължава на следващата страница				

Таблица 2 – продължение от предишната страница

Стъпки на цикъл преди връщане на "Hello, World!"	WebFrame++	Node.JS Express	Сравнение м/у WebFrame++ и Express	
2048	0m0.228s	0m0.027s	-0.0034m	-88.16%
4096	0m0.228s	0m0.034s	-0.0032m	-85.09%
8192	0m0.225s	0m0.029s	-0.0033m	-87.11%
16384	0m0.225s	0m0.032s	-0.0032m	-85.78%
32768	0m0.225s	0m0.031s	-0.0032m	-86.22%
65536	0m0.225s	0m0.034s	-0.0032m	-84.89%
131072	0m0.226s	0m0.038s	-0.0031m	-83.19%
262144	0m0.227s	0m0.053s	-0.0029m	-76.65%
524288	0m0.228s	0m0.075s	-0.0026m	-67.11%
1048576	0m0.229s	0m0.119s	-0.0018m	-48.03%
2097152	0m0.231s	0m0.213s	-0.0003m	-7.79%
4194304	0m0.239s	0m0.387s	0.0025m	61.92%
8388608	0m0.242s	0m0.770s	0.0088m	218.18%
16777216	0m0.258s	0m1.556s	0.0216m	503.10%
33554432	0m0.290s	0m3.006s	0.0453m	936.55%
67108864	0m0.360s	0m5.873s	0.0919m	1531.39%
134217728	0m0.475s	0m11.954s	0.1913m	2416.63%
268435456	0m0.716s	0m23.419s	0.3784m	3170.81%
536870912	0m1.217s	0m46.988s	0.7629m	3760.97%
1073741824	0m2.199s	1m34.091s	1.5315m	4178.81%
Най-голямото изоставане на WebFrame++			-0.0034m	-88.65%
Най-големият аванс на WebFrame++			1.5315m	4178.81%
Най-малкото изоставане на WebFrame++			-0.0003m	-7.79%
Най-малкият аванс на WebFrame++			0.0025m	61.92%

3.2.2 Сума на N произволни числа

Вторият тест се представя с псевдо код по следния начин:

Algorithm 2

```
1: procedure Request
2:   N  $\leftarrow$  number
3:   sum  $\leftarrow$  global reference
4: begin:
5:   i := 0
6: loop:
7:   if i == N then return "Hello, World!"
8:   if i < N then
9:     i := i + 1
10:    num  $\leftarrow$  random generated number
11:    sum := (sum + num) mod  $10^9$ 
12:    goto loop.
```

Използвани са вградените в езиците генератори на „произволни“ числа с оглед това, че генерирането на „произволни“ числа е една от най-бавните операции в езиците.

3.2.2.1 Сравнение с Flask

Таблица 3: Сравнение м/у WebFrame++ и Flask (тежки операции)

Брой събрани произволни числа преди връщане на "Hello, World!"	WebFrame++	Flask	Сравнение м/у WebFrame++ и Flask	
2	0m0.228s	0m0.269s	0.0007m	17.98%
4	0m0.224s	0m0.271s	0.0008m	20.98%
8	0m0.236s	0m0.268s	0.0005m	13.56%
16	0m0.226s	0m0.269s	0.0007m	19.03%
32	0m0.224s	0m0.272s	0.0008m	21.43%
64	0m0.226s	0m0.271s	0.0008m	19.91%
128	0m0.225s	0m0.271s	0.0008m	20.44%
256	0m0.227s	0m0.271s	0.0007m	19.38%
512	0m0.225s	0m0.271s	0.0008m	20.44%
1024	0m0.226s	0m0.282s	0.0009m	24.78%
Продължава на следващата страница				

Таблица 3 – продължение от предишната страница

Брой събрани произволни числа преди връщане на "Hello, World!"	WebFrame++	Flask	Сравнение м/у WebFrame++ и Flask	
2048	0m0.226s	0m0.277s	0.0009m	22.57%
4096	0m0.224s	0m0.293s	0.0012m	30.80%
8192	0m0.227s	0m0.306s	0.0013m	34.80%
16384	0m0.227s	0m0.323s	0.0016m	42.29%
32768	0m0.224s	0m0.382s	0.0026m	70.54%
65536	0m0.236s	0m0.517s	0.0047m	119.07%
131072	0m0.227s	0m0.755s	0.0088m	232.60%
262144	0m0.230s	0m1.299s	0.0178m	464.78%
524288	0m0.249s	0m2.265s	0.0336m	809.64%
1048576	0m0.250s	0m4.255s	0.0668m	1602.00%
2097152	0m0.262s	0m8.349s	0.1348m	3086.64%
4194304	0m0.288s	0m16.366s	0.2680m	5582.64%
8388608	0m0.351s	0m32.125s	0.5296m	9052.42%
16777216	0m0.455s	1m3.665s	1.0535m	13892.31%
33554432	0m0.693s	2m8.230s	2.1256m	18403.61%
67108864	0m1.139s	4m19.225s	4.3014m	22659.00%
134217728	0m2.056s	8m35.322s	8.5544m	24964.30%
268435456	0m3.888s	8m38.944s	8.5843m	13247.33%
536870912	0m7.481s	17m19.063s	17.1930m	13789.36%
1073741824	0m14.729s	33m54.976s	33.6708m	13716.12%
Най-голямото изоставане на WebFrame++			NaN	NaN
Най-големият аванс на WebFrame++			33.6708m	24964.30%
Най-малкото изоставане на WebFrame++			NaN	NaN
Най-малкият аванс на WebFrame++			0.0005m	13.56%

3.2.2.2 Сравнение с Express

Таблица 4: Сравнение м/у WebFrame++ и Express
(тежки операции)

Брой събрани произволни числа преди връщане на "Hello, World!"	WebFrame++	Flask	Сравнение м/у WebFrame++ и Express
Продължава на следващата страница			

Таблица 4 – продължение от предишната страница

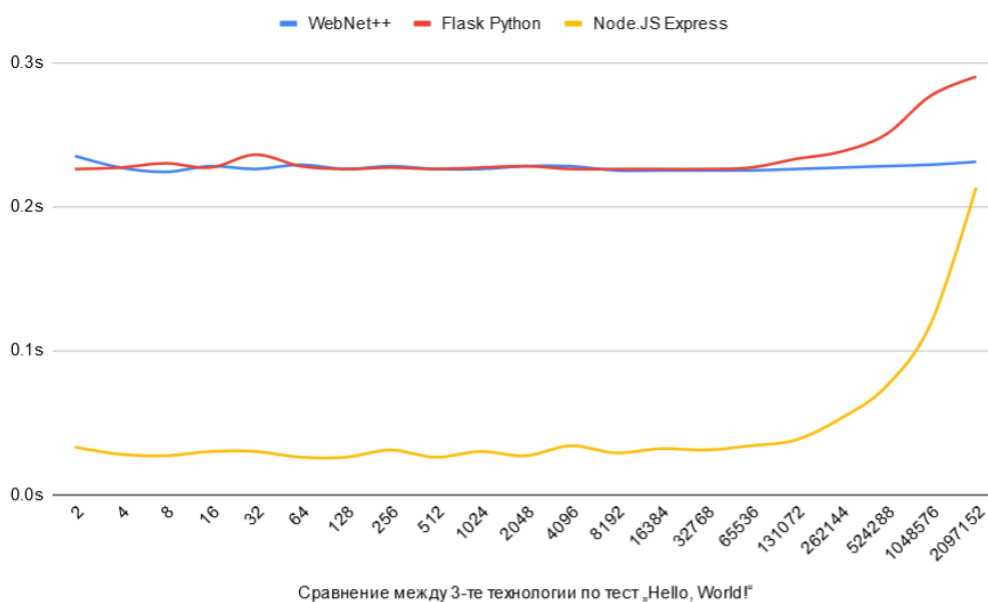
Брой събрани произволни числа преди връщане на "Hello, World!"	WebFrame++	Express	Сравнение м/у WebFrame++ и Express	
2	0m0.228s	0m0.036s	-0.0032m	-84.21%
4	0m0.224s	0m0.024s	-0.0033m	-89.29%
8	0m0.236s	0m0.024s	-0.0035m	-89.83%
16	0m0.226s	0m0.029s	-0.0033m	-87.17%
32	0m0.224s	0m0.024s	-0.0033m	-89.29%
64	0m0.226s	0m0.024s	-0.0034m	-89.38%
128	0m0.225s	0m0.024s	-0.0034m	-89.33%
256	0m0.227s	0m0.024s	-0.0034m	-89.43%
512	0m0.225s	0m0.025s	-0.0033m	-88.89%
1024	0m0.226s	0m0.024s	-0.0034m	-89.38%
2048	0m0.226s	0m0.025s	-0.0034m	-88.94%
4096	0m0.224s	0m0.028s	-0.0033m	-87.50%
8192	0m0.227s	0m0.025s	-0.0034m	-88.99%
16384	0m0.227s	0m0.029s	-0.0033m	-87.22%
32768	0m0.224s	0m0.028s	-0.0033m	-87.50%
65536	0m0.236s	0m0.032s	-0.0034m	-86.44%
131072	0m0.227s	0m0.042s	-0.0031m	-81.50%
262144	0m0.230s	0m0.053s	-0.0030m	-76.96%
524288	0m0.249s	0m0.081s	-0.0028m	-67.47%
1048576	0m0.250s	0m0.135s	-0.0019m	-46.00%
2097152	0m0.262s	0m0.223s	-0.0007m	-14.89%
4194304	0m0.288s	0m0.422s	0.0022m	46.53%
8388608	0m0.351s	0m0.792s	0.0074m	125.64%
16777216	0m0.455s	0m1.573s	0.0186m	245.71%
33554432	0m0.693s	0m3.361s	0.0445m	384.99%
67108864	0m1.139s	0m6.472s	0.0889m	468.22%
134217728	0m2.056s	0m12.895s	0.1807m	527.19%
268435456	0m3.888s	0m24.948s	0.3510m	541.67%
536870912	0m7.481s	0m49.342s	0.6977m	559.56%
1073741824	0m14.729s	1m39.670s	1.4157m	576.69%
Най-голямото изоставане на WebFrame++			-0.0035m	-89.83%
Най-големият аванс на WebFrame++			1.4157m	576.69%
Най-малкото изоставане на WebFrame++			-0.0007m	-14.89%
Продължава на следващата страница				

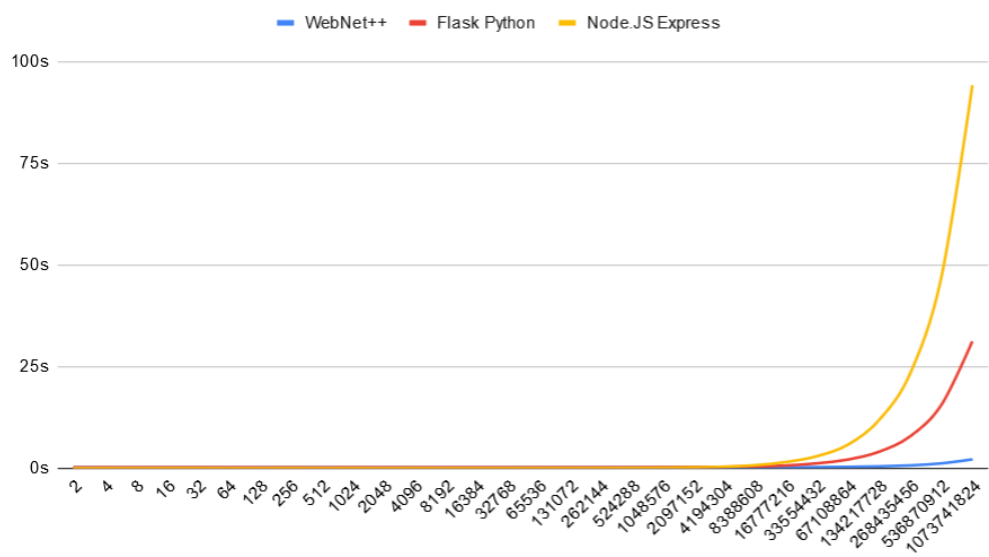
Таблица 4 – продължение от предишната страница

Брой събрани произволни числа преди връщане на "Hello, World!"	WebFrame++	Express	Сравнение м/у WebFrame++ и Express	
Най-малкият аванс на WebFrame++			0.0022m	46.53%

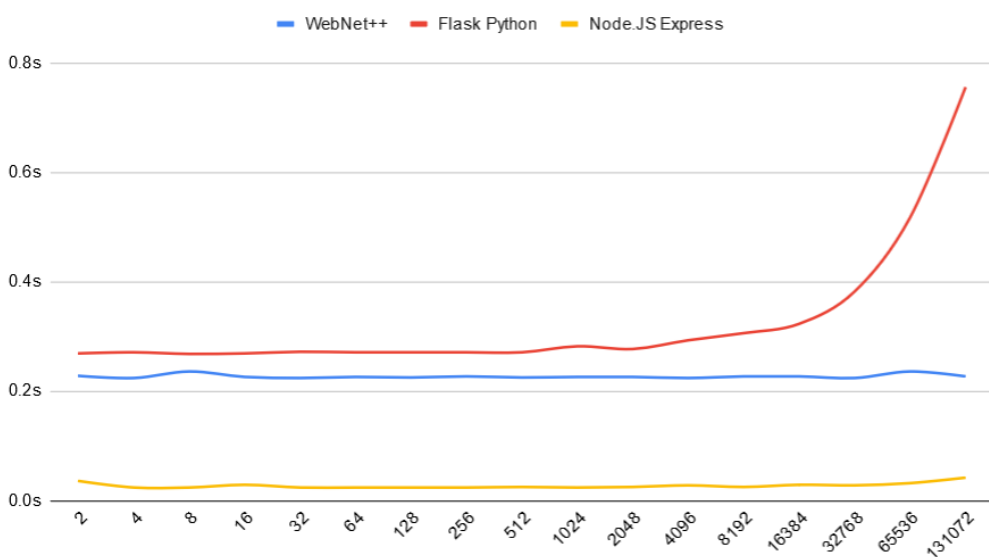
3.2.3 Извод

WebFrame++[8] надминава в пъти Flask, но спрямо Express при „по-леките“ (като брой и необходима процесорна мощ) операции изостава с времето.

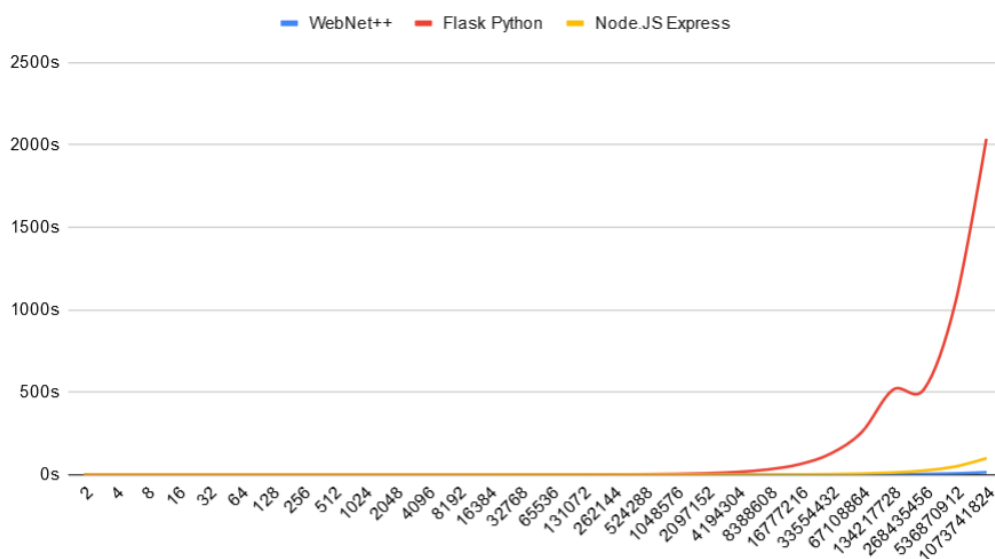




Сравнение между 3-те технологии по тест „Hello, World!“



Сравнение между 3-те технологии по тест „Сума на n произволни числа“



Сравнение между 3-те технологии по тест „Сума на n произволни числа“

В сравнението между WebFrame++[8] и Express лесно се забелязва, че най-бавната операция в библиотеката е свързването с клиента, което в момента не е оптимизирано напълно. Това означава, че в бъдеще с допълнително развиване на тази част от библиотеката, тя може да подобри представянето дори на Node.JS. Оказва се, че в момента WebFrame++[8] е една от най-бързите библиотеки за работа с големи данни, които позволяват работа със сървър. Това е очевидно от горепоказаните тестове и в пъти по-добрите резултати на библиотеката спрямо световните лидери.

4 Технологии

- C++17[1] е стандартът, на който е писана библиотеката
- Flask[3] като инструмент за изграждане на уеб сървър на Python
- Express[5] като инструмент за изграждане на уеб сървър на Node.JS

5 Бъдеща разработка и разпространение

Оптимизиране на скоростта

Предстои да се подобри скоростта при свързването с клиента. Така тази част от библиотеката ще може да подобри представянето дори и на Node.JS.

По-дълбоки изследвания на представянето

Предстои да бъдат пренаписани някои по-сложни уеб разработки с помощта на WebFrame++ и да се сравни, както времето за реакция на сървъра, така и използваните ресурси, като памет и процесорна мощ.

Още проучвания и сравнения

Предстои да бъде изследвана ефективността на библиотеката с други конкуренти като Spring Framework 5 [6] за Java, Laravel [4] и Symfony [7] за PHP и други.

CppCon [2]

Започната е подготовка за лекция на световната конференция CppCon, която цели да събере C++ програмисти от целия свят, които да разменят опит помежду си.

6 Благодарности

Благодарности за помощта при оформянето на документацията:

- Надежда Цачева
- Димо Чанев

Благодаря още на:

- Ученическият институт на БАН
- Българска академия на науките

Литература

- [1] C++17. <https://en.cppreference.com/>.
- [2] Cppcon. <https://cppcon.org/>.
- [3] Flask. <https://github.com/pallets/flask>.
- [4] Laravel. <https://laravel.com/>.
- [5] Node.js express. <https://expressjs.com/>.
- [6] Spring framework 5. <https://spring.io/>.

- [7] Symfony. <https://symfony.com/>.
- [8] Webframe++. <https://github.com/Alex-Tsvetanov/Web-CPP>, <https://github.com/WebFrame/Core>.
- [9] Анонимна анкета сред 65 ученици от 10. и 11. клас на СМГ. <https://forms.gle/YRgRrUwagnNwwS7a8>.