

Podrobná uživatelská dokumentace

Tato dokumentace obsahuje podrobný návod ke všem programům, které obsahuje příloha práce. Popis systémových požadavků, přístupu k programátorské dokumentaci a návod jak spustit ukázkou projektu obsahuje dokument `uživatelská dokumentace.pdf`.

Všechny cesty uvedené v tomto souboru jsou relativní k cestě složky, která obsahuje PDF soubor této dokumentace.

Nativní knihovna

Pokud neplánujete používat skripty v jazyce Python, můžete sestavit nativní knihovnu samostatně pomocí následujících příkazů. Vygenerované řešení nebude využívat SIMD instrukcí. Pokud těchto instrukcí chcete využít, vygenerujte řešení pomocí skriptu `src/scripts/buildProject.py`.

```
1 mkdir src/cmakeBuild
2 cd src/cmakeBuild
3 cmake -DCMAKE_BUILD_TYPE=Release ../index
4 cmake --build . --config Release
```

Řešení bude vytvořeno ve složce `src/cmakeBuild`. V každém systému vypadají soubory řešení jinak. Např. při použití Windows s Visual Studiem je řešením `.sln` soubor a projekty jsou `.vcxproj` soubory. Pro spuštění projektů je doporučena konfigurace `Release`. Řešení obsahuje dva projekty.

- recallTable* - Postaví HNSW index nové implementace a vypíše tabulku závislosti přesnosti na parametru vyhledávání `efsearch`. Více o konfiguraci tohoto projektu v kapitole `JSON konfigurace`.

Parametr	Význam	Výchozí hodnota
První	Cesta ke konfiguračnímu souboru typu JSON.	<code>src/config/config.json</code>

- datasetToText* - Vypíše textový popis datové kolekce do souboru. Slouží pro ověření konzistence mezi binárními a HDF5 soubory. Název datového souboru je prvním parametrem programu. Výchozí hodnotou je `angular-small`.

JSON Konfigurace

Pro změnu sestavovaných konfigurací programy `recallTable.cpp` a `recallTable.py` upravte soubor `src/config/config.json`. V souboru je jediný JSON objekt, který obsahuje dva povinné klíče.

```
1 {
2     "datasets": [
3         {
4             "name": "angular-small",
5             "angular": true,
6             "dim": 25,
7             "k": 10,
```

```

8         "testCount": 200,
9         "trainCount": 20000,
10        "seed": 104
11    }
12 ],
13 "index": [
14     {
15         "dataset": "angular-small",
16         "efConstruction": 200,
17         "efSearch": [10, 15, 20, 40, 80, 120, 200],
18         "mMax": 16,
19         "seed": 200,
20         "SIMD": "best",
21         "template": "prefetching"
22     }
23 ]
24 }

```

Klíč `datasets` je pole objektů, kde každý objekt popisuje jeden datový soubor.

Klíč	Typ hodnoty	Význam
name	string	Unikátní název souboru sloužící k identifikaci.
angular	boolean	Pokud je nastaven na <code>true</code> , využívá soubor kosinusové vzdálenosti. Jinak využívá Eukleidovské vzdálenosti.
dim	int	Počet dimenzí prostoru.
k	int	Počet hledaných nejbližších sousedů dotazovaného prvku.
testCount	int	Počet dotazů.
trainCount	int	Počet prvků použitých k sestavení indexu.
seed	int	Nastavení generátoru náhodných čísel.

Klíč `index` je pole objektů, kde každý objekt popisuje jednu konfiguraci indexu.

Klíč	Typ hodnoty	Význam
dataset	string	Identifikace datového souboru. Odpovídá klíči <code>name</code> v souboru <code>src/config/debugDatasets.json</code> .
efConstruction	int	Počet uvažovaných sousedů při vytváření nových hran v indexu.
efSearch	array	Pole hodnot parametru vyhledávání <code>ef_{search}</code> .
mMax	int	Maximální povolený počet sousedů jednoho prvku v indexu na vrstvě vyšší než vrstva 0.
seed	int	Nastavení generátoru náhodných úrovní v indexu.
SIMD	string	Upřednostňovaný typ SIMD instrukcí. Možnosti jsou <code>avx</code> , <code>avx512</code> , <code>best</code> , <code>null</code> , a <code>sse</code> .*
template	string	Šablona indexu. Možnosti jsou <code>Heuristic</code> , <code>Naive</code> , <code>NoBitArray</code> a <code>Prefetching</code> .

* Zvolením hodnoty `best` zvolíte nejmodernější dostupné SIMD rozšíření. Hodnotou `null` zakážete použití SIMD instrukcí.

Šablony nové implementace

Šablona	Metoda výběru sousedů	Seznam navštívených vrcholů	Asynchronní přístup do paměti
Heuristic	Heuristika	Bitové pole	Ne
Naive	Naivní algoritmus	Bitové pole	Ne
NoBitArray	Heuristika	Obyčejné pole	Při výpočtu vzdáleností Při načítání dat seznamu navštívených vrcholů
Prefetching	Heuristika	Bitové pole	Při výpočtu vzdáleností

Sestavení rozhraní v jazyce Python

Pokud chcete využít skriptů jazyka Python, musíte vygenerovat správné virtuální prostředí. Pokud jste již alespoň jednou spustili ukázkou pomocí skriptu `RUNME.py`, můžete tuto sekci přeskočit.

Spusťte skript `src/scripts/buildProject.py` pomocí interpretu Python verze 3.9. Tento skript vygeneruje virtuální prostředí, stáhne potřebné balíčky a zkompileje knihovnu nové implementace indexu HNSW.

OS Windows

```
1 | py -3.9 .\src\scripts\buildProject.py
```

```
1 python3.9 ./src/scripts/buildProject.py
```

Virtuální prostředí

Pokud není uvedeno jinak, skripty uvnitř složky `src/scripts` vždy spouštějte pomocí vygenerovaného virtuálního prostředí. Prostředí aktivujete pomocí aktivačního skriptu. Výběr skriptu závisí na použitém OS a interpretu.

OS	Interpret	Cesta k aktivačnímu skriptu
Linux	*	<code>./venv/bin/activate</code>
Windows	Batch	<code>.\venv\Scripts\activate.bat</code>
Windows	Powershell	<code>.\venv\Scripts\Activate.ps1</code>

* Pokud používáte OS Linux specifikujte zvolený interpret při volání aktivačního skriptu.

```
1 source ./venv/bin/activate
```

Srovnání implementací

Skript `src/runBenchmarks.py` spustí srovnání implementací v jednom nebo více Docker kontejnerech, vypočítá výkonnostní metriku, vygeneruje webovou stránku s výsledky a otevře ji v nové kartě internetového prohlížeče. Kód vygenerované stránky lze poté najít ve složce `src/website` a můžete ji opětovně zobrazit otevřením souboru `index.html`.

Před spuštěním se ujistěte, že je služba Docker zapnutá.

Parametr, zkratka	Význam
<code>--algoDefPaths, -a</code>	Vyžadován. Seznam cest ke konfiguračním souborům oddělených mezerami. O konfiguraci se více dočtete v kapitole konfigurace srovnání .
<code>--datasets, -d</code>	Vyžadován*. Seznam datových souborů oddělených mezerami.
<code>--datasetsPath, -p</code>	Vyžadován*. Cesta k textovému souboru se seznamem datových souborů.
<code>--force, -f</code>	Spustí již provedená měření znovu.-
<code>--runs, -r</code>	Počet opakování měření. Výchozí hodnota je 1.
<code>--workers, -w</code>	Počet paralelně spuštěných Docker kontejnerů. Výchozí hodnota je 1.

* Pouze jeden z parametrů označených hvězdičkou by měl být uveden.

Datové soubory využitě ke srovnání nejsou ty samé, které jsou využívány k debugování nativní knihovny. Jejich seznam najdete v kapitole [Testované datové soubory](#). Příklad spuštění:

```
1 python runBenchmarks.py -a ..\config\noBit.yaml -f -p ..\config\datasets.txt  
-r 5 -w 2
```

Konfigurace srovnání

Výběr implementací ke srovnání a jejich parametrů zprostředkovávají konfigurační soubory ve formátu YAML. Příklad takového souboru je `src/config/algos.yaml`. Ve složce `src/config` se nacházejí předem vytvořené konfigurace.

Následuje příklad konfigurace. Komentáře označené znakem `#` popisují jednotlivé klíče.

```
1 algos: # Povinný klíč.  
2   # Povinné pole názvů porovnávaných implementací.  
3   - original  
4   - new-prefetch  
5 build: # Povinný klíč.  
6   # Povinné pole objektů konfigurace stavby.  
7   # Pro každou konfiguraci musí být uvedeny hodnoty parametrů efConstruction  
8   a mMax.  
9   - efConstruction: 50  
10  mMax: 4  
11  - efConstruction: 100  
12  mMax: 8  
13 efSearch: [10, 12, 14, 16, 18, 20, 25, 30, 40, 80] # Povinné pole hodnot  
14 parametru vyhledávání efSearch.
```

Definované implementace popisuje následující tabulka.

Název implementace	Druh implementace	Šablona	SIMD rozšíření
original	Původní hnsplib		Nejmodernější dostupné
new-avx	Nová	Heuristic	AVX
new-avx-512	Nová	Heuristic	AVX-512
new-heuristic	Nová	Heuristic	Nejmodernější dostupné
new-naive	Nová	Naive	Nejmodernější dostupné
new-no-bit	Nová	NoBitArray	Nejmodernější dostupné
new-no-simd	Nová	Heuristic	Žádné
new-prefetch	Nová	Prefetching	Nejmodernější dostupné
new-sse	Nová	Heuristic	SSE

Popis šablon obsahuje kapitola `Šablony nové implementace`.

Testované datové soubory

Pro srovnání implementací je možno využít následujících datových souborů.

Název	Dimenze	Počet prvků při stavbě	Dotazy	Metrika
fashion-mnist-784-euclidean	784	60 000	10 000	Eukleidovská vzdálenost
glove-25-angular	25	1 183 514	10 000	Kosinusová vzdálenost
glove-50-angular	50	1 183 514	10 000	Kosinusová vzdálenost
glove-100-angular	100	1 183 514	10 000	Kosinusová vzdálenost
glove-200-angular	200	1 183 514	10 000	Kosinusová vzdálenost
lastfm-64-dot	64	292 385	50 000	Kosinusová vzdálenost
mnist-784-euclidean	784	60 000	10 000	Eukleidovská vzdálenost
nytimes-16-angular	16	290 000	10 000	Kosinusová vzdálenost
nytimes-256-angular	256	290 000	10 000	Kosinusová vzdálenost
random-s-100-angular	100	100 000	10 000	Kosinusová vzdálenost
random-s-100-euclidean	100	100 000	10 000	Eukleidovská vzdálenost
random-xs-20-angular	20	10 000	10 000	Kosinusová vzdálenost
random-xs-20-euclidean	20	10 000	10 000	Eukleidovská vzdálenost
sift-128-euclidean	128	1 000 000	10 000	Eukleidovská vzdálenost

Pro spuštění srovnání nad více soubory lze využít textového formátu, kde každý řádek reprezentuje jeden datový soubor. Řádky, které začínají znakem `#` jsou ignorovány. Příklad:

```
1 # glove-25-angular
2 nytimes-256-angular
3 sift-128-euclidean
```

Seznam dalších skriptů

Následuje seznam dalších skriptů ve složce `src/scripts`.

Název skriptu	Stručný popis skriptu
<code>buildProject</code>	Vytvoří virtuální prostředí, nativní C++ řešení a jeho Python rozhraní.
<code>clean</code>	Odstraní vygenerované soubory a vrátí projekt do původního stavu.
<code>datasetGenerator</code>	Vygeneruje datové soubory pro debugování.
<code>datasetToText</code>	Převede datový soubor do textového formátu.
<code>formatCMakeTemplates</code>	Vygeneruje <code>CMakeLists.txt</code> .
<code>generateTables</code>	Vygeneruje LaTeX tabulky podobné těm, které jsou v bakalářské práci.
<code>latexTable</code>	Vygeneruje LaTeX tabulku na základě výsledků srovnání.
<code>runRecallTable</code>	Postaví nový index a zobrazí tabulku závislosti přesnosti na parametru vyhledávání <code>ef_{search}</code> .
<code>SIMDCapability</code>	Zobrazí SIMD rozšíření instrukční sady procesoru, která jsou k dispozici.

Podrobný popis skriptů

U každého skriptu je uveden jeho účel, parametry a příklad spuštění. Pokud skript obsahuje alespoň jeden parametr, pak použitím parametru `--help` nebo `-h` zobrazíte nápovědu v anglickém jazyce.

`buildProject`

Tento skript lze spustit bez virtuálního prostředí.

Vytvoří virtuální prostředí interpretu Python, stáhne potřebné softwarové balíčky, vygeneruje nativní C++ řešení pro knihovnu nového indexu, vytvoří rozhraní v jazyce Python pro nový index a otestuje funkčnost tohoto indexu spuštěním skriptu `runRecallTable`.

Parametr, zkratka	Význam
<code>--clean, -c</code>	Vrátí projekt do původního stavu před jeho opětovným sestavením.
<code>--cleanResults, -r</code>	Pokud je <code>--clean</code> nastaven, odstraní naměřené výsledky.
<code>--ignorePythonVersion, -i</code>	Umožňuje spustit skript s libovolnou verzí interpretu Python. Skript poté nemusí fungovat správně.

Příklad spuštění:

```
1 | py -3.9 buildProject.py --clean --cleanResults
```

clean

Tento skript lze spustit bez virtuálního prostředí.

Odstraní datové soubory pro debugování, C++ nativní řešení a Python rozhraní. Pokud je spuštěn mimo virtuální prostředí, pak odstraní toto prostředí. Naměřené výsledky odstraněny nebudou, pokud o to uživatel nepožádá.

Parametr, zkratka	Význam
--results, -r	Odstraní naměřené výsledky srovnání, vygenerované grafy a tabulky.

Příklad spuštění:

```
1 | python clean.py --results
```

datasetGenerator

Vygeneruje datové soubory pro debugování. O konfiguraci tohoto skriptu se více dočtete výše v kapitole `JSON konfigurace`.

Parametr, zkratka	Význam	Výchozí hodnota
--config, -c	Cesta ke konfiguračnímu souboru typu JSON.	<code>src/config/config.json</code>

Příklad spuštění:

```
1 | python datasetGenerator.py
```

datasetToText

Převede vybraný datový soubor ze složky `src/data` do textového formátu. Výstupní textový soubor zapíše pod jménem datového souboru do stejné složky.

Parametr, zkratka	Význam
--name, -n	Název datového souboru bez přípony. Pokud není uveden, výchozím souborem je <code>angular-small</code> .

Příklad spuštění:

```
1 | python datasetToText --name euclidean-medium
```


formatCMakeTemplates

Vygeneruje soubor `src/index/CMakeLists.txt` a doplní do něj správnou definici maker tak, aby došlo pouze ke kompilaci těch funkcí, pro které je k dispozici vhodné SIMD rozšíření instrukční sady procesoru.

Příklad spuštění:

```
1 python formatCMakeTemplates.py
```

generateTables

Vygeneruje LaTeX tabulky podobné těm, které jsou v bakalářské práci, ale pouze v případě, že jsou pro ně dostupné naměřené výsledky. Tyto výsledky lze získat spuštěním následujících příkazů. Avšak tato měření mohou trvat více než 12 hodin.

```
1 python runBenchmarks.py -a ..\config\heuristic.yaml ..\config\naive.yaml -d lastfm-64-dot -r 5
2 python runBenchmarks.py -a ..\config\heuristic.yaml ..\config\prefetch.yaml -d glove-50-angular -r 5
3 python runBenchmarks.py -a ..\config\original.yaml ..\config\prefetch.yaml -d sift-128-euclidean -r 5
4 python generateTables.py
```

Vygenerované tabulky jsou dostupné ve složce `src/figures`.

latexTable

Vygeneruje jednu LaTeX tabulku na základě výsledků srovnání implementací.

Parametr, zkratka	Význam
--algorithms, -a	Vyžadován. Seznam implementací oddělený mezerami.
--dataset, -d	Vyžadován. Název datového souboru.
--label, -la	Identifikátor tabulky.
--legend, -le	Názvy implementací v tabulce. Pokud není uveden, budou použity původní názvy.
--output, -o	Vyžadován. Cesta k výstupnímu souboru.
--percent, -p	Přidá do tabulky sloupec s procentuálním rozdílem časů stavby.
--recompute, -r	Znovu vypočítá výkonnostní metriky z naměřených výsledků. Tato operace může trvat více než 10 minut.

Příklad spuštění:

```
1 python latexTable.py -a new-prefetch original -d sift-128-euclidean -le "Nová impl." "Původní impl." -o ..\figures\table.tex -p
```

runRecallTable

Postaví index nové implementace a vyhledá v něm nejbližší sousedy s různými hodnotami parametru vyhledávání ef_{search} . Poté vypíše tabulku závislosti přesnosti vyhledávání na tomto parametru. Konfigurace tohoto skriptu se nachází v souboru `src/config/recallTable.json` a více se o ní dočtete v kapitole `Konfigurace programů recallTable`.

Příklad spuštění:

```
1 | python runRecallTable.py
```

SIMDCapability

Zobrazí SIMD rozšíření instrukční sady procesoru, která jsou k dispozici. Využívána ostatními skripty pro vygenerování správných maker v jazyce C++.

Příklad spuštění:

```
1 | python SIMDCapability.py
```