

Цель работы

Целью работы является изучение и сравнение стратегий кэширования **LRU** и **LFU**, реализованных в итеративной и рекурсивной формах. Необходимо разработать эмулятор кэша, провести оценку производительности и эффективности алгоритмов по основным метрикам, а также автоматизировать сбор и визуализацию результатов для последующего анализа.

Ход работы

Отчет общей эффективности на рисунках 1-2

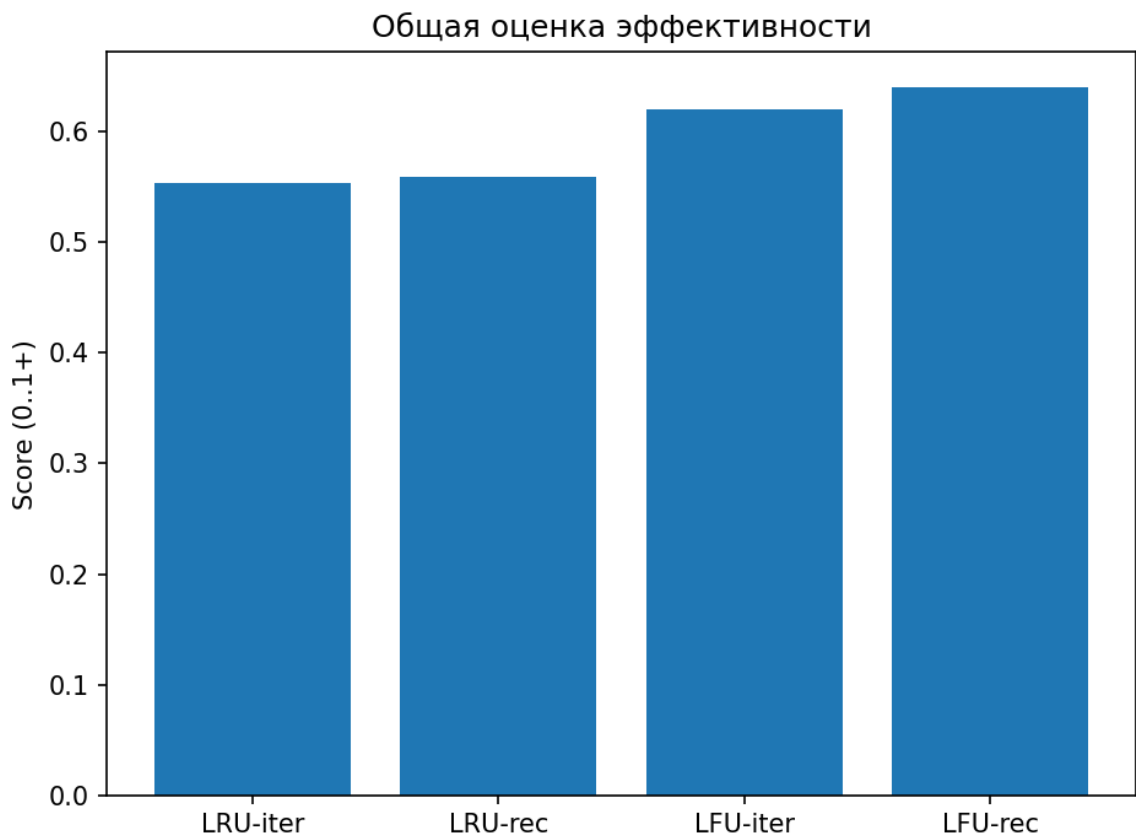


Рисунок 1

build > efficiency_score.csv							
Algo	Impl	Score	Stability_score	Hit_rate	Avg_ns	Memory_eff	
LRU	iter	0.55	36.44	6.3	30.29	100	
LRU	rec	0.56	85.17	6.3	206.53	100	
LFU	iter	0.62	53.14	20.28	52.39	100	
LFU	rec	0.64	87.72	25.61	203.79	100	

Рисунок 2

Зависимость времени и производительности от размера кэша представлена на рисунке 3.

build > scalability_extended.csv

	Size ▾	Algo ▾	Impl ▾	Elapsed_ns ▾	Avg_ns ▾	Ops_per_sec ▾	Hit_rate ▾	Useful_eviction	Harmful_eviction	Eviction_efficiency
	16	LRU	iter	793260	52.86	18919400	0.04	1016	3213	24.02
	16	LFU	iter	1003201	66.84	14960100	0.94	1019	3153	24.42
	16	LRU	rec	577991	38.51	25965800	0.04	1016	3213	24.02
	16	LFU	rec	596906	39.77	25143000	2.13	1019	3292	23.64
	32	LRU	iter	740871	49.34	20268000	0.07	1010	2875	26
	32	LFU	iter	969336	64.55	15491000	1.64	1019	2750	27.04
	32	LRU	rec	1002740	66.78	14975000	0.07	1010	2875	26
	32	LFU	rec	998050	66.47	15045300	3.76	1019	3005	25.32
	64	LRU	iter	651340	43.33	23078600	0.11	994	2174	31.38
	64	LFU	iter	867632	57.72	17325300	2.53	1017	1914	34.7
	64	LRU	rec	1900079	126.4	7911250	0.11	994	2174	31.38
	64	LFU	rec	1928513	128.29	7794610	6.39	1014	2404	29.67
	128	LRU	iter	581487	38.6	25906000	0.47	941	1064	46.93
	128	LFU	iter	743105	49.33	20271700	5.58	1015	454	69.09
	128	LRU	rec	3642747	241.82	4135340	0.47	941	1064	46.93
	128	LFU	rec	3633280	241.19	4146120	12.63	1004	1405	41.68
	256	LRU	iter	604401	39.95	25029700	10.71	800	206	79.52
	256	LFU	iter	863415	57.07	17521100	21.17	905	40	95.77
	256	LRU	rec	5856453	387.13	2583130	10.71	800	206	79.52
	256	LFU	rec	5854669	387.01	2583920	25	921	236	79.6
	512	LRU	iter	619891	40.63	24610800	49.33	613	4	99.35
	512	LFU	iter	1066210	69.89	14308600	49.52	618	3	99.52
	512	LRU	rec	11672479	765.11	1307010	49.33	613	4	99.35
	512	LFU	rec	16310151	1069.1	935368	50.27	645	0	100
	1024	LRU	iter	575596	37.11	26949500	78.43	140	0	100
	1024	LFU	iter	1193653	76.95	12995400	78.43	139	0	100
	1024	LRU	rec	20279390	1307.34	764915	78.43	140	0	100
	1024	LFU	rec	45528204	2935.03	340712	78.53	146	0	100

Рисунок 3

Экономическая эффективность (ROI) представлена на рисунке 4

build > roi.csv

	Algo ▾	Impl ▾	Roi ▾	Perf_score ▾	Resource ▾	Impl_cost ▾	Maint_cost ▾
	LRU	iter	407758	1837720	1.01	2	1.5
	LRU	rec	67157.9	304996	2.04	1.5	1
	LFU	iter	781397	3915100	1.51	2	1.5
	LFU	rec	228682	1267020	3.04	1.5	1
*							

Рисунок 4

```
build > warmup.csv
```

Step	Hit_rate
0	20.44
1	15.44
2	12.78
3	9.42
4	10.28
5	8.63
6	5.99
7	7.23
8	7.44
9	5.1
10	6.22
11	4.04
12	3.58
13	2.6
14	1.59
15	1.61
16	1.09
17	1.43
18	0.45
19	1

Рисунок 5

График времени от размера кэша представлен на рисунке 6.

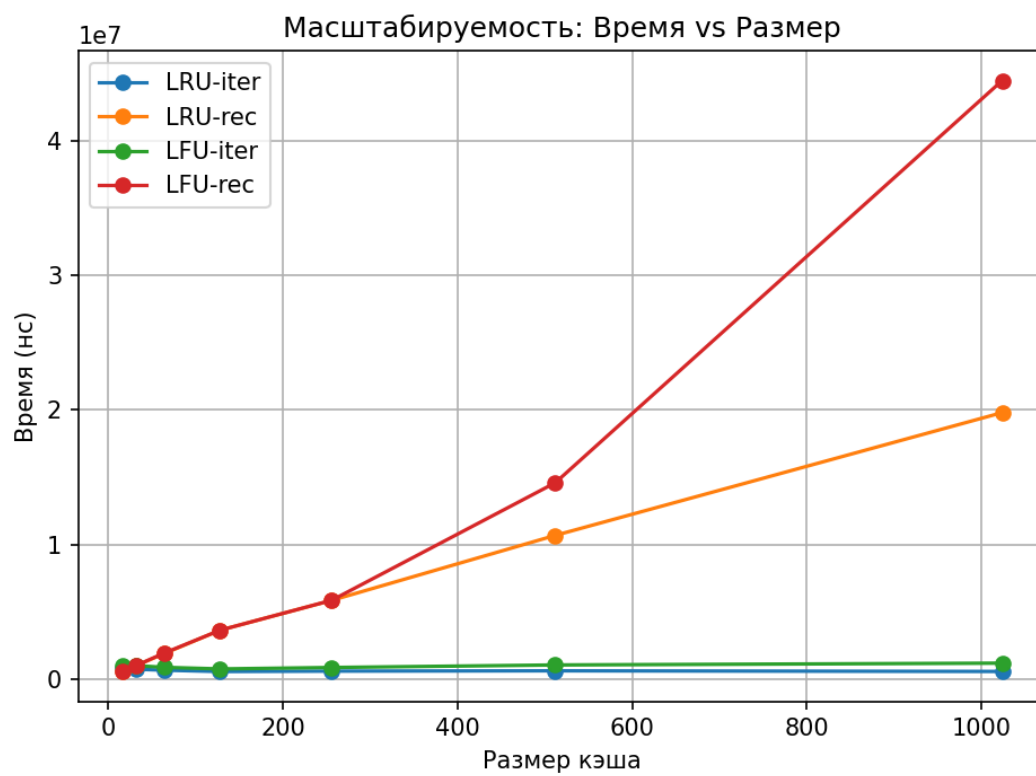


Рисунок 6

Вывод программы в VS Code на рисунке 4

```
[gregory@PP sam7]$ rm -rf build && find . -type f -name '*.png' -delete && cmake -B build && cmake --build build && cd ./build && ./app && cd .. && python3 plot_metrics_ext.py .
--- Проверка корректности LRU/LFU ---
LRU (iter) Test: OK
LFU (iter) Test: OK

CSV-файлы сохранены:
- results_extended.csv
- scalability_extended.csv
- stability.csv
- efficiency_score.csv
- roi.csv
- algorithm_efficiency.csv
- warmup.csv
QSocketNotifier: Can only be used with threads started with QThread
Сохранены графики:
- scalability_time_ext.png
- scalability_hit_ext.png
- eviction_eff_bar.png
- efficiency_score.png
- roi_bar.png
```

Рисунок 4

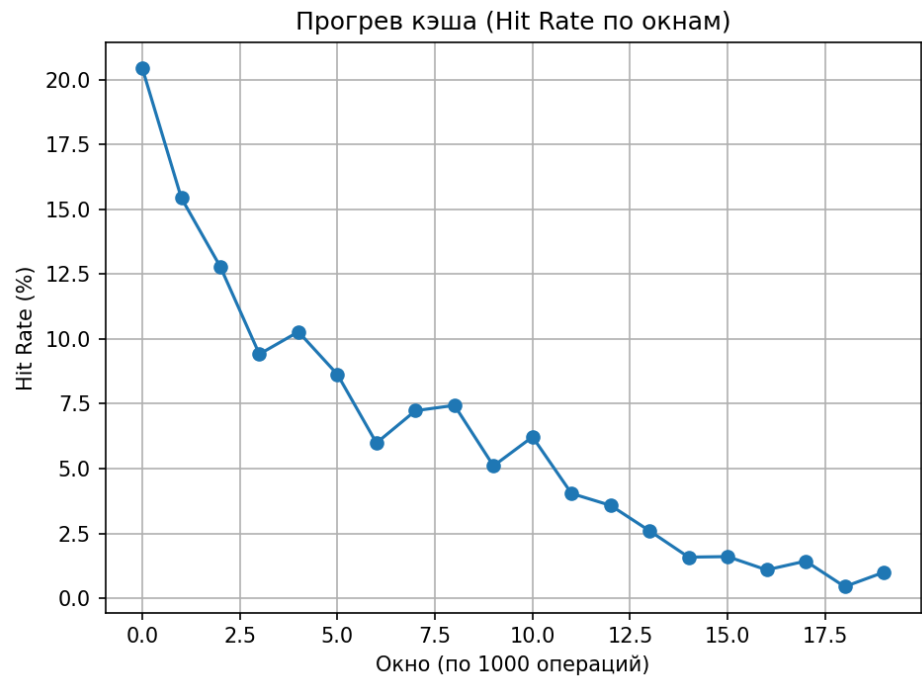


Рисунок 5

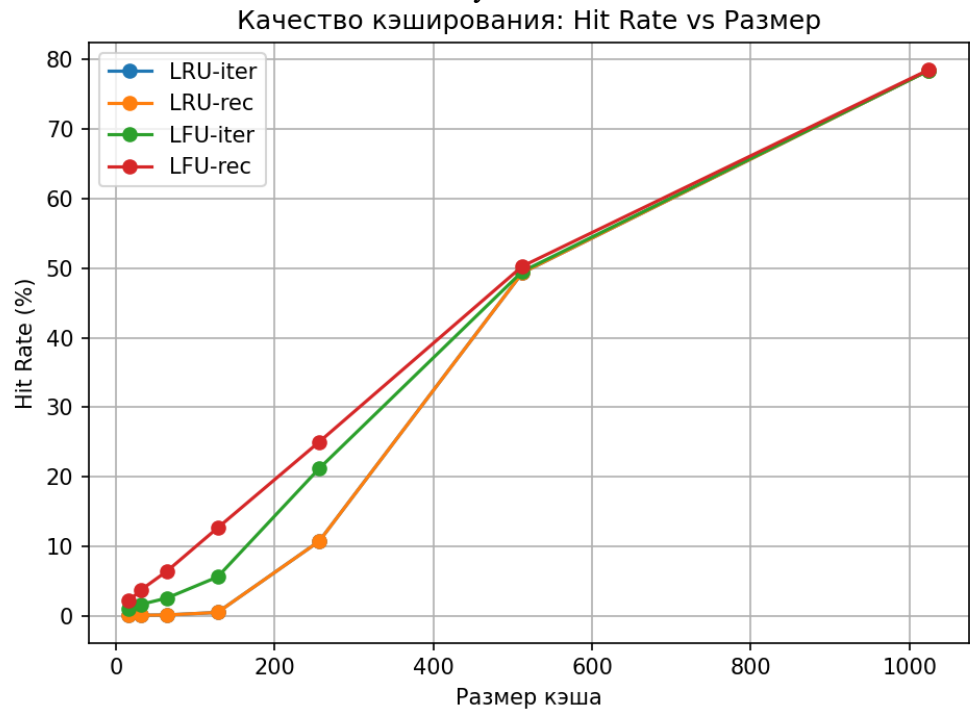


Рисунок 6

Вывод

В ходе работы были реализованы и исследованы стратегии кэширования LRU и LFU в итеративной и рекурсивной формах. Проведён анализ их производительности по ключевым метрикам — *Hit Rate*, *Average Access Time*, *Operations per Second* и другим.

Результаты показали, что **итеративные реализации работают быстрее**, тогда как **рекурсивные — более наглядны и структурированы**, но менее эффективны по времени..

Поставленные цели и задачи работы выполнены полностью.