

설명서

실행 방법

1. Docker Desktop 실행
2. docker-compose.yml 있는 디렉토리로 이동
3. `docker compose build` 로 프로젝트 빌드
4. `docker compose up -d` 로 프로젝트 실행
나중에 프로젝트 종료시 `docker compose down`
5. `curl -X POST --data-binary @{trace_data.csv} localhost:3000/trace` 로 서버에 원하는 trace 데이터 추가
6. <http://localhost:3000/index.html> 메인 UI 접근

화면 설명

Trace Info 화면

Trace Info	
id	name
13	7/31/2022, 1:41:55 PM
14	7/31/2022, 1:41:55 PM_1
15	7/31/2022, 1:41:56 PM
16	7/31/2022, 1:41:56 PM_1
17	7/31/2022, 1:41:57 PM
18	7/31/2022, 1:41:57 PM_1
19	7/31/2022, 1:41:57 PM_2
20	7/31/2022, 1:41:58 PM
21	8/1/2022, 12:22:06 AM
22	8/1/2022, 12:22:06 AM_1
<div><< >> <input type="text"/> Go 2/3 SELECT</div>	

- 시각화에 사용할 trace 선택하는 화면

이동:

원하는 trace를 선택하고 오른쪽 아래 SELECT 버튼 ⇒ **Trace 보기 화면** 으로 이동

Trace 보기 화면

Trace 보기

그냥 보기

특정 tx랑 겹친 경우 보기

- 그냥 보기: trace 전체 보기
- 특정 tx랑 겹친 경우 보기: 특정 트랜잭션이랑 겹치는 트랜잭션들만 모아 보기

이동:

그냥보기 버튼 ⇒ 그냥 보기(그래프) 화면 으로 이동

특정 tx랑 겹친 경우 보기 버튼 ⇒ tx 고르기 화면 으로 이동

tx 고르기 화면

tx 고르기

SQL 실행

id	traceld	requestsid	gen	fin	txSource	txType	txId
69178	11	GC	1370875	2080583	GC	WRITE	
69179	11	GC	1370875	2080583	GC	WRITE	
69180	11	GC	1370875	2080583	GC	WRITE	
69181	11	GC	1370875	2080583	GC	WRITE	
69323	11	GC	1370875	2088447	GC	WRITE	
69324	11	GC	1370875	2088447	GC	WRITE	
69325	11	GC	1370875	2088447	GC	WRITE	
69326	11	GC	1370875	2088447	GC	WRITE	
66696	11	101976	1970031	1991559	USERIO	WRITE	0
66693	11	101982	1970140	1991559	USERIO	WRITE	0

<< >>

Go 1/667

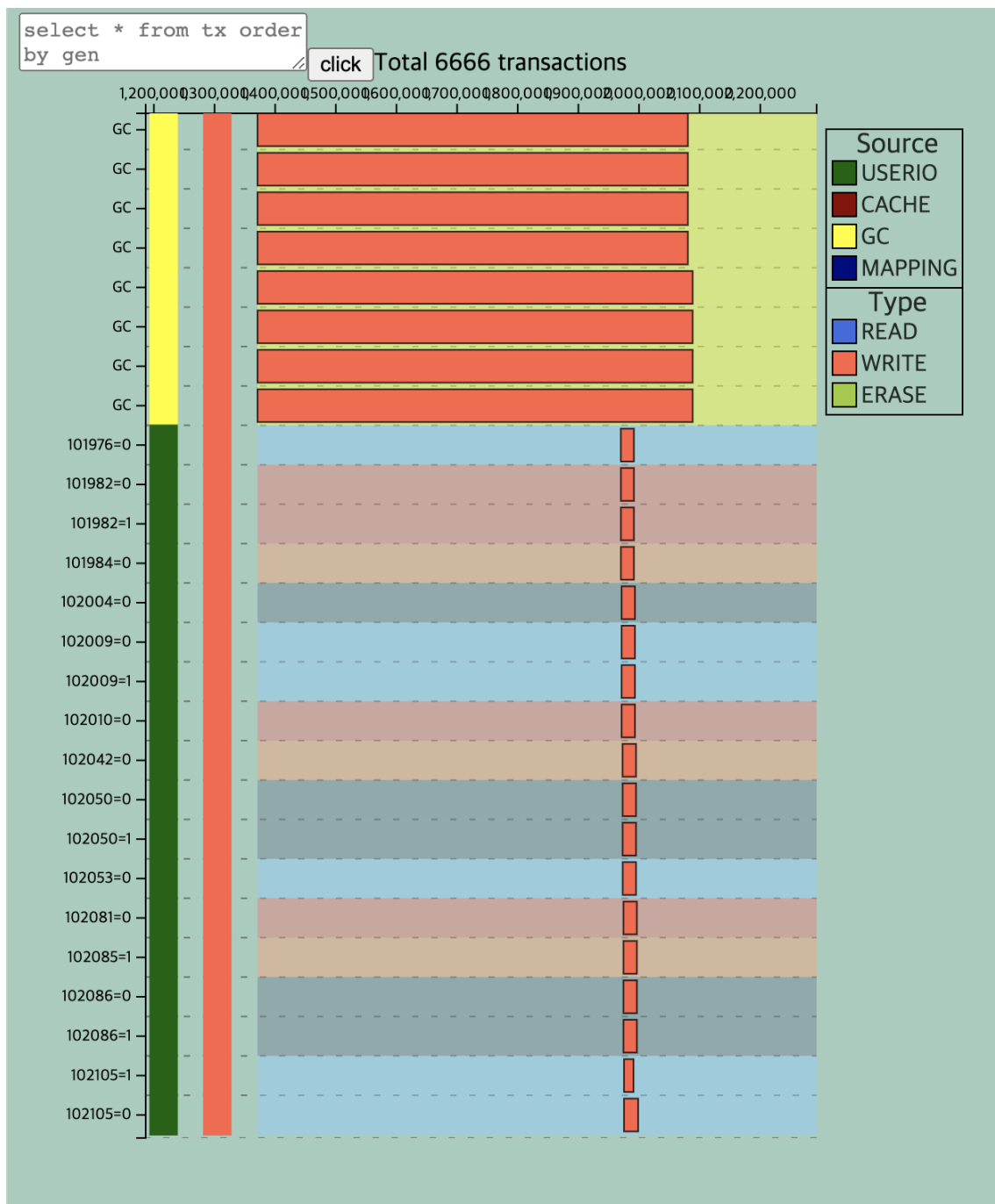
SELECT

- 타킷 고르는 화면
- 왼쪽 텍스트 창: sql로 원하는 tx 검색 (자세한 이용방법은 SQL 입력창 관련 참고)

이동:

원하는 tx 선택 후 SELECT 버튼 ⇒ tx 고르기 화면 으로 이동

그냥 보기(그래프) 화면



- 시각화 화면
- 왼쪽 텍스트 창: sql로 특정 트랜잭션들만 보기 가능 (SQL 입력창 관련 참고)
- 그래프: 마우스로 줌이랑 이동 가능
 - x축: 시간
 - y축: 트랜잭션들
 - 왼쪽 열(노란색, 초록색): 트랜잭션들 Source
 - 그 다음 열(주황색): 트랜잭션들 Type
 - 바 그래프: 트랜잭션들 시작과 끝 표시
 - 배경색: 같은 requestId를 가진 트랜잭션들은 같은 배경색

타깃 보기(그래프) 화면



- 타깃 트랜잭션이랑 겹치는 트랜잭션들 시각화 화면
- 그냥 보기(그래프) 화면이랑 비슷

차이점

- sql 입력 창이 없음
- source랑 type을 보여주는 열 사이에 원하는 target인지 나타내는 열 추가

SQL 입력창 관련

- mysql 사용
- 테이블 이름은 tx 사용
 - 예시) select * from tx

tx 테이블 구조

column	id	requestId	gen	fin	txSource	txType	txId
type	int	text	int	int	text	text	int
비고	primary key				USERIO, CACHE, GC, MAPPING 값들만 존재	READ, WRITE, ERASE 값들만 존 재	txSource가 USERIO가 아닌 경우 NULL

Docker Compose 관련

- `docker compose build`: 프로젝트 빌드(처음에 한번만 실행하면 됨)
- `docker compose up`: 프로젝트 실행
 - `docker compose up -d`: 프로젝트 백그라운드로 실행
- `docker compose down`: 프로젝트 종료

DB 관련

`mysql -h localhost -P 3306 --protocol=tcp -u root -D records`: db cli에 직접 접근

서버 REST API 관련

```
curl localhost:3000/trace
```

현재 저장된 trace 데이터 JSON 형태로 보기

```
curl -X POST --data-binary @{{trace_data.csv}} localhost:3000/trace
```

새로운 trace 데이터를 저장. Trace 이름은 현재 시간을 사용.

```
curl -X POST --data-binary @{{trace_data.csv}} localhost:3000/trace/{{traceName}}
```

새로운 trace 데이터를 저장. Trace 데이터 이름은 traceName을 사용 .

```
curl -X DELETE localhost:3000/trace/{{traceId}}
```

주어진 traceId를 가진 trace 데이터 삭제