프로젝트 2: Recovery System

2019년 5월 1일

제출 기한: 5월 17일 금요일 23시 59분

개요

프로젝트 2에서는 지난 프로젝트 1에서 만든 Wikipedia 문서 검색 시스템의 데이터베이스를 업데이트하는 시스템을 만들어본다. 우선 시스템은 주어진 schedule을 처리하면서 failure에 대비하여 log를 생성한다. Schedule을 처리하는 중에 failure를 만나면 앞서 생성한 log를 이용해 데이터베이스를 recover한다.

요구 사항

1 제공된 schedule 파일에 있는 read/write command(SQL syntax로 주어짐)를 순서대로 처리하고 데이터베이스 업데이트.

2 데이터베이스 업데이트와 동시에 recovery에 필요한 log record를 생성해 log 파일에 기록.

3 Schedule 파일에서 "system failure - recover"를 만나면 log-based recovery 수행.

4 Recovery 이후의 데이터베이스 상태를 확인하기 위해 검색 질의(query)를 이용해 답을 확인할 것이므로 프로젝트 1의 기능(pagerank)이 수행 가능해야 함.

상세 스펙

1 **schedule** 파일(**prj2.sched**) 포맷

1.1 파일 포맷 (**입력 파일 형식**)

항목	Command	설명
1	<Transaction ID> [SQL]	해당 transaction의 SQL 처리.
2	<Transaction ID> commit	해당 transaction을 commit.
3	<Transaction ID> rollback	해당 transaction을 rollback.
4	checkpoint	체크포인트 수행.
5	system failure - recover	Failure 발생 후 recovery 수행
6	search <keyword>	주어진 키워드를 이용해 검색 수행

1.2 파일 예시

```
<T1> DELETE FROM wiki WHERE id = '339712';
<T2> UPDATE wiki SET text = 'Les Humphries Singers Sing Sang Song Germany 1976' WHERE id = '6241635';
<T1> DELETE FROM link WHERE id_from = '339712';
<T1> DELETE FROM link WHERE id_to = '339712';
<T2> rollback
system failure - recover
<T3> DELETE FROM wiki WHERE id = '7649292';
<T3> DELETE FROM link WHERE id_from = '7649292';
<T3> DELETE FROM link WHERE id_to = '7649292';
<T3> commit
checkpoint
system failure - recover
search germany
```

2 **log** 파일(**prj2.log**) 포맷

2.1 파일 포맷 (다른 형태도 허용.)

포맷	설명
<Transaction ID> start	해당 transaction 시작.
<Transaction ID>, <Table>.<key>.<column>, <oldValue>, <newValue>	해당 table의 특정 값을 변경.
<Transaction ID> commit	해당 transaction commit.
<Transaction ID> abort	해당 transaction rollback.
checkpoint <Transaction ID>,...,<Transaction ID>	체크포인트에서의 active transaction.
recover <line number>	Failure후 recovery 수행.

3 Schedule 파일의 실행

3.1 실행 방법

- **"-run"** 명령을 내려 schedule 파일의 command를 처리하면서 log 파일에 log를 기록.

3.2 실행 예시

```
i-love-d@tabase:~$ python main.py
building tables…
ready to search
2017-12345> -run prj2.sched
2017-12345>
```

4 Schedule 파일의 처리

4.1 1.1 항에 제시된 테이블 상세 설명.

항목	Command	설명
1	<Transaction ID> [SQL]	해당 transaction의 SQL 처리.
2	<Transaction ID> commit	해당 transaction을 commit.
3	<Transaction ID> rollback	해당 transaction을 rollback.
4	checkpoint	체크포인트 수행.
5	system failure - recover	Failure 발생 후 recovery 수행
6	search <keyword>	주어진 키워드를 이용해 검색 수행

항목 1. SQL 형식은 아래 세 종류로 한정하며, WHERE절에 논리 연산자(AND, OR)는 사용되지 않음.

```
UPDATE wiki SET {title | text} = <value> WHERE id = <value>;
DELETE FROM wiki WHERE id = <value>;
DELETE FROM link WHERE {id_from | id_to} = <value>;
```

항목 5.Recovery 수행 후 redo, undo한 transaction ID 리스트를 파일에 저장.

항목 6. 프로젝트 1에서 만든 Wikipedia 문서 검색 시스템(pagerank)을 이용해 검색을 수행하고 결과를 파일에 저장.

4.2 "system failure - recover"를 만날 경우,

- log-based recovery 수행.
- Recovery시 수행되는 redo, undo 과정도 log에 기록.
- Commit된 transaction들의 update가 반영된 DB 기준으로 TF-IDF와 PageRank 재계산.
- log에 "recover <line number>" 기록.
  - ✓ Line number는 schedule 파일에서 failure가 일어난 행의 번호.
  - ✓ Line number는 1부터 시작.
- Recovery가 끝나면 log에 "checkpoint" 기록. (= active transaction이 없는 상태)
- redo와 undo된 transaction id를 파일에 저장. (자세한 내용은 아래 5.2항 참조)

4.3 "search <검색 질의(query)>"를 만날 경우,

- 검색 질의(query)에 대해 프로젝트 1과 같은 방식으로 결과를 구한 후 **파일에 저장**. (자세한 내용은 아래 5.3항 참조)
- Search command는 active한 transaction이 없을 때에만 등장한다고 가정.

4.4 힌트: 각 테이블에 해당 레코드가 유효한지 나타내는 Boolean(binary) 데이터타입의 column(delete flag)을 추가하여 DELETE문 처리에 이용할 수도 있음. 단, delete flag를 이용하여 구현할 경우 retrieval 측면에서도 고려되어야 할 것임.

5 출력 파일

5.1 Log 파일

- 파일명은 **prj2.log**로 설정.
- log는 생성된 순서대로 한 log 파일에 기록.
- 포맷에 대한 자세한 사항은 2.1항 참조.

5.2 Recovery 수행 결과 파일

- 파일명은 **recovery.txt**로 설정.
- 여러 번 recovery가 일어날 경우 새로운 결과를 기존 결과 뒤에 덧붙임(append).

- 파일 포맷

```
recover <line number>
redo <Transaction ID>,…,<Transaction ID>
undo <Transaction ID>,…,<Transaction ID>

…
```

- 파일 예시

```
recover 4
redo <T2>
undo <T1>, <T3>
recover 5
redo
undo
```

5.3 Search 수행 결과 파일

- 파일명은 **search.txt**로 설정.
- 여러 번 검색이 일어날 경우 새로운 결과를 기존 결과 뒤에 덧붙임(append).
- 파일 포맷

```
search <line number>
query <검색 질의(query)>
<문서id>, <문서title>, <TF-IDF>, <PageRank>
<문서id>, <문서title>, <TF-IDF>, <PageRank>
…
```

(상위 10개 문서의 id, title, TF-IDF score, PageRank score 출력)

- 파일 예시

```
search 8
query program
22398341, Fire-safe_polymers, 0.1579736481364444, 0.00015617679212868969
11993020, Pizza_Corner, 0.1534747598430004, 0.00015617679212868969
48174128, Lori_Weitzner, 0.15109672465364796, 0.00015617679212868969
6684154, Symphony_Services_International, 0.14267337499095695, 0.0005627992299881567
11741801, Surf_II, 0.13702271198107013, 0.00015617679212868969
33599991, Riot_grrrl, 0.13549384882472837, 0.0015027541995585045
39546115, Graham_Gold, 0.11957890446819681, 0.00015617679212868969
11761630, Solid_Ground_(Seattle), 0.1169726651079252, 0.00015617679212868969
41422654, Golden_Boy_(novel), 0.11681532582408478, 0.00015617679212868969
3235632, Proud_Mary, 0.10621294103089528, 0.00015617679212868969
search 9
query lands
5080244, De_Lacy, 0.12132564179582231, 0.00015617679212868969
41235373, Kingdom_of_Hungary_(1000%E2%80%931301), 0.12062883295541851, 0.00015617679212868969
35086251, Gary_M._Feinman, 0.11863338817046515, 0.00015617679212868969
31285205, Antoni_Pawe%C5%82_Su%C5%82kowski, 0.11022541394078135, 0.00015617679212868969
2833267, Coastal_management, 0.10265534460296169, 0.00015617679212868969
25693804, Richard_Basset, 0.1003271319452092, 0.00015617679212868969
15982010, Battle_of_the_Eurymedon, 0.07819046544219166, 0.00015617679212868969
```

개발 환경(PRJ 1 과 동일)

1 개발 언어: **Python3**

2 데이터베이스: mysql (pymysql 패키지 이용)

IP	Port	Id	Password
s.snu.ac.kr	3306	ADB_학번 (ex. ADB2016_12345)	ID와 동일

제출

1 main.py를 포함하는 소스코드와 requirements.txt 파일

- NLTK, pymysql 이외의 package를 쓴다면 requirements.txt로 명시
- 핵심적인 구현을 외부 package를 이용하면 감점 사유가 될 수 있음.
- Linux 환경에서 main.py가 실행되지 않거나 입/출력 포맷을 지키지 않은 경우 감점.

2 리포트

- 파일 형식: pdf, docx, hwp만 허용.
- 분량: 10페이지 이내. 필수 작성 항목은 다음과 같음. 필요에 따라 항목 추가 가능.

필수 작성 항목
개발 환경
구현 기능 소개 (간략하게)
log 파일 생성 과정 및 코드 설명
recovery 과정에 사용한 SQL문 설명
프로그램 실행 예시 (화면 캡쳐 포함)
평가 및 결론

3 제출 방식

- main.py를 포함하는 소스코드, 리포트를 하나의 파일로 압축하여 메일로 제출.
- 압축 파일 명: PRJ2_학번_이름.zip (ex. PRJ2_2016-12345_홍길동.zip)
- E-mail 제목: [ADB]PRJ2_학번_이름 (ex. [ADB]PRJ2_2016-12345_홍길동)
- E-mail 주소: lecture@europa.snu.ac.kr
- 제출 기한: **5월 17일 금요일 23시 59분**

평가 항목

항목	배점	비고
Main.py 파일의 정상 실행 여부	5	바로 실행 불가 또는 포맷 미준수 시 감점
프로그램 실행 입/출력 포맷 준수 여부	5	
schedule 파일의 command 정상 처리 여부	10	별도의 테스트셋으로 채점
prj2.log 정상 생성 여부	10	
recovery.txt 정상 생성 여부	20	
search.txt 정상 생성 여부	20	
schedule 처리 (run) 소요 시간	20	장시간 소요 시 감점
리포트	10	
총점	100	

- 쿼드 코어 i7,ram 16gb 환경에서 실험 예정
- 총 실행 시간(5개 이내의 쿼리)이 5분이 넘어가면 시간 항목 0점, 15분이 넘게 걸리는 경우 전체 항목 0점 처리.
- PRJ1에서 실행이 너무 오래 걸리는 사람은 레퍼런스 코드를 사용하여 구현.

주의 사항

1 소스코드 카피, 보고서 표절 시 해당 프로젝트 전체 점수 0점

2 과제 제출 기한을 넘길 시, E-mail 제출 기한을 기준으로 매 24시간 마다 10%씩 감점.
(ex. 1일 지연 시 10%, 2일 지연 시 20%, 3일 지연 시 30%)

3 제출 기한으로부터 3일이 넘으면 제출 불가.

4 E-mail 반송, 첨부 파일 누락, 파일 실행 불가 시 마지막 제출일을 기준으로 위 감점 기준 적용.

참고 자료

1 Mysql
https://www.mysql.com

2 PyMySql
https://pymysql.readthedocs.io/en/latest/

3 MySQL Workbench
https://www.mysql.com/products/workbench/

4 NLTK Tokenizer
https://www.nltk.org/api/nltk.tokenize.html