DB Internals Project

120050009 Deepanjan Kundu
120050082 Roshan R

1 Problem Statement

Use PF layer to simulate RAID01 disk system over n disks. Consider a workload W (consisting of a sequence of read/write operations) and get the DB performance. Next, assume that we need to take a snapshot of the database just before we start W. Use the copy-on-write technique for the snapshot. After the snapshot, we resume the workload W, but also initiate in parallel, the backup of the DB from the snapshot. Simulate this situation now, and get DB performance for the workload W when the backup is also going on.

2 Data Structures

2.1 Work Item

A work item contains

- Type - Workload or Backup
- Operation kind - Read or Write
- The page number on which to perform the operation
- Pointer to a buffer from/in which data is to be read/written
- Timestamp

2.2 Snapshot

This will be responsible for maintaining a point-in-time copy of the underlying database using the copy-on-write technique. It will contain

- A flag to enable/disable snapshotting
- The name of the file in which the snapshot is stored
- Set of page numbers currently in snapshot
- Last accessed page number of the snapshot
- A map which maps the original page numbers in RAID system to actual page numbers in snapshot file of the stored pages
- A reference to the underlying RAID system

2.3 RAID01

The RAID system is built upon the PF layer. Files are used to represent disks and each block in a disk is considered as a page. It contains

- The number of disks to be simulated
- List containing file names corresponding to each of the disks
- List containing last accessed page numbers for each disk
- List containing flags which indicate which of the disks are available for read/write
- The current time point in the simulation
- A queue in which the past work items will be buffered in order to determine concurrency

3 Algorithms and Implementation Details

3.1 Generate test files containing workload and backup

1. The workload and backup items are generated by a seperate program which generates them alternatively
2. The program is run with minor changes to generate different test cases for analysis
3. From this, a script is used to extract the workload items alone for analysis

```
for(int i=0;i<200;i++)
{
    if(i<WORKLOAD) cout<<0<<" "<<rand()%2<<" "
                        <<rand()%200<<" "<<t<<endl;
    if(i<200) cout<<1<<" "<<0<<" "<<i<<" "<<t<<endl;
    t+=(rand()%3 > 1)*RATE;
}
```

3.2 Processing of work item by Snapshot Software

1. Check if snapshotting is enabled. If not, directly forward to RAID system.
2. For backup items, if the necessary page is present in the snapshot map, then fetch the page from the snapshot file. Else, forward to RAID system.

```
map<int, int>::iterator iter = pageNumbers.find(item.pageNumber);
if(iter != pageNumbers.end()) {
    char *page;
    int fd = PF_OpenFile(fileName);
    PF_GetThisPage(fd, iter->second, &page);
    memcpy(item.buffer, page, PF_PAGE_SIZE);
    PF_UnfixPage(fd, iter->second, FALSE);
```

```
    PF_CloseFile(fd);

    read_num2 += 1;
    if(iter->second != previous + 1) {
        seek_num2 += 1;
    }
    previous = iter->second;
} else {
    raidSystem.add_workItem(item);
}
```

3. For workload items, if the operation kind is Write and the page is not present in snapshot file,
   (a) Read the page from RAID system
   (b) Write it to snapshot file
   (c) Add an entry in the snapshot map

```
raidSystem.add_workItem(readItem);
int fd = PF_OpenFile(fileName);
PF_AllocPage(fd, &pno, &page);
memcpy(page, buf, PF_PAGE_SIZE);
PF_UnfixPage(fd, pno, TRUE);
PF_CloseFile(fd);
pageNumbers[item.pageNumber] = pno;
```

4. Forward the item to the RAID system.

3.3 Processing of work item in RAID system

1. Execute the work item
2. Get disk number and page number within disk

```
int disk_num = w.pageNumber % n_disk;
int page_num = w.pageNumber / n_disk;
```

3. If timestamp of item is more than the current time point in the RAID system, flush all the disk flags
4. If the item can be run without any seeks, then return
5. If the disks which the item needs are available, push it onto the queue and mark the disks as not available
6. Else flush the queue and disk flags, advance the simulation by one timestep and push the item onto the queue.

Algorithm 1 Execute work item in RAID 0+1 system

1: $\mathcal{I} \leftarrow$ Work Item
2: $\mathcal{W} \leftarrow$ Writes
3: $\mathcal{R} \leftarrow$ Reads
4: $\mathcal{N} \leftarrow$ Number of disks
5: $diskNumber \leftarrow$ PageNumber$(\mathcal{I})\%(N/2)$
6: $pageNumber \leftarrow$ PageNumber$(\mathcal{I})/(N/2)$
7: **if** type$(\mathcal{I})$ = Read **then**
8: $\mathcal{R} \leftarrow \mathcal{R} + 1$
9: Read from $pageNumber$ in file on $diskNumber$
10: **else**
11: $\mathcal{W} \leftarrow \mathcal{W} + 2$
12: Write to $pageNumber$ in file on $diskNumber$
13: Write to $pageNumber$ in file on $diskNumber + (N/2)$

3.4 Making a RAID system on top of PF layer

The RAID system is simulated as n files representing disks and pages within files representing disk blocks. Each file will have a suitable number of pages allocated and written. No new pages will be added. This will simulate limited disk space.

Any operation involves first obtaining the correct file and the correct page within the file. Then the file is opened and the page is loaded onto the PF layer's buffer. Then depending on the operation type, contents are transferred from the work item's buffer to the PF layer's buffer or from the PF layer's buffer to the work item's buffer. Then the page is unfixed and the file is closed.

3.5 Time calculation

The RAID system tries to perform as many parallel operations as possible without reordering the work items. For simplicity, it has been assumed that the time taken for this is approximately 1 seek time since the read and write time is negligible for a page size of 4KB. Therefore, the time has been expressed in units of the seek time.

4 Performance analysis

To analyse the performance of the system, the number of items in the workload and the item rate of the workload and backup were varied. This gave rise to nine test cases :

Test Case 0 : Equal items in workload as backup, Low item rate
Test Case 1 : Equal items in workload as backup, Medium item rate
Test Case 2 : Equal items in workload as backup, High item rate
Test Case 3 : Twice as many items in workload as backup, Low item rate
Test Case 4 : Twice as many items in workload as backup, Medium item rate
Test Case 5 : Twice as many items in workload as backup, High item rate
Test Case 6 : Half as many items in workload as backup, Low item rate
Test Case 7 : Half as many items in workload as backup, Medium item rate
Test Case 8 : Half as many items in workload as backup, High item rate

The above analysis was repeated again after removing all backup items.

4.1 Results

In case of only workload(no backup),

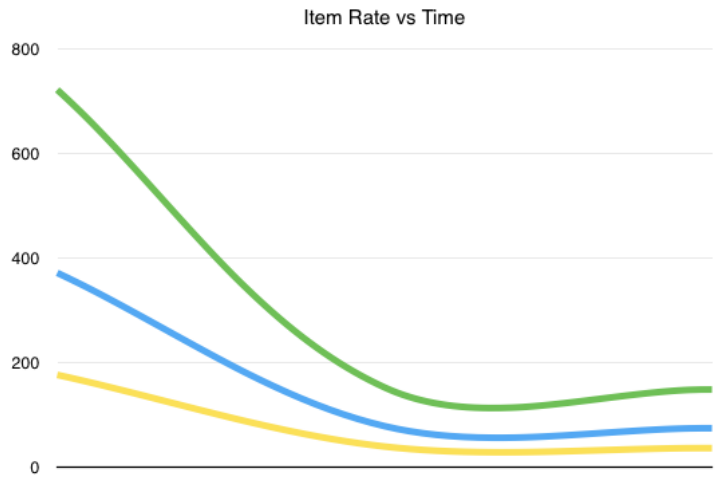
	Raid			
Case	Seeks	Reads	Writes	Time
0	39	96	208	371
1	74	296	208	78
2	74	296	208	74
3	72	198	404	721
4	148	198	404	152
5	148	198	404	148
6	17	49	102	176
7	36	49	102	39
8	36	49	102	36

When workload and backup are combined,

	Raid			Snapshot			
Case	Seeks	Reads	Writes	Seeks	Reads	Writes	Time
0	112	339	208	62	39	82	371
1	154	339	208	62	39	82	157
2	156	339	208	62	39	82	156
3	171	484	404	62	39	125	721
4	260	484	404	62	39	125	263
5	262	484	404	63	39	125	262
6	57	267	102	31	25	43	370
7	80	267	102	31	25	43	83
8	82	267	102	31	25	43	82

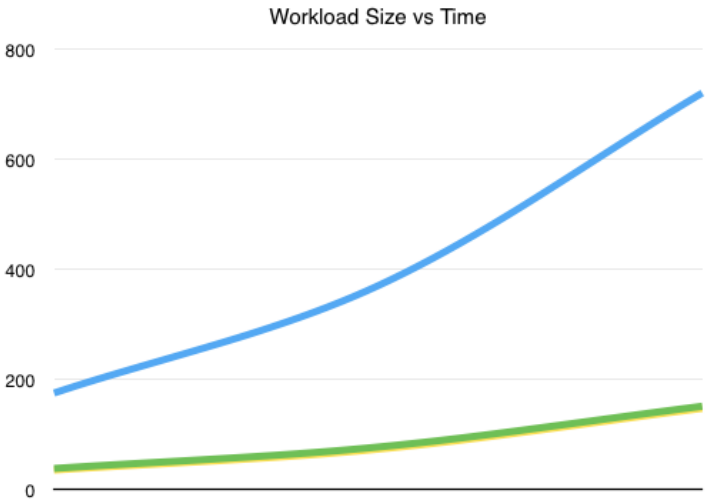
4.2 Effect of Item Rate

The time taken decreases with increase in item rate at first and then flattens out.
Reason : At first, the items were not coming at fast enough rate compared to the seek time. So, the time taken is approximately equal to the time at which last item was executed. As the item rate increases, the seek time becomes the bottleneck and so the graph flattens out.



4.3 Effect of Workload Size

The time taken increases with increase in item rate.
Reason : Given that the work items arrive at the same rate, for a larger workload size, it takes more time.



4.4 Workload performance with/without Backup

The time taken to execute the workload with backup is always greater than or equal to the time taken to execute the workload without backup.

With decreasing item rate, the workload performance with backup approaches that of without backup. This is because enough time is given to process the workload and the backup.