

ソフトウェア工学

2021年10月1日

川島英之

自己紹介

データ集約型科学

すばる望遠鏡：300GB/夜

↓ HTAP(SQL+TX)を提供せよ！

データシステム

上と下からの変革要求

↑ 我々を活用するなら再設計せよ！

先進的ハードウェア

不揮発メモリ、メニーコアCPU

- 2005
 - 博士号、安西研@慶大
- 2007～2013
 - 筑波大計算科学研究センター
- 2018～
 - 慶大SFC
- テーマ
 - データ集約型科学
 - データシステム



Data Platform Laboratory

共同研究状況

- Cybozu Lab: 星野喬
- キオクシア: 長谷川揚平・安福健太
- KDDI研究所: 斉藤和広
- 三菱東京UFJ: 尾根田倫太郎
- 科研費基盤B: 藤原靖宏@NTT、松谷宏紀@矢上
- NEDO: 神林飛志@ノーチラステクノロジーズ
- ヤフー研究所
- NTT研究所

授業の狙い

- Learn the following through DBMS kernel implementation
 - System programming techniques
 - Software engineering
- Why DBMS?
 - I have some expertise
 - DBMS is a sort of a treasure box

Materials

- Textbook for “System Programming”
 - syspro-kiso-text-jp.pdf
- UNIX
 - <http://cns-guide.sfc.keio.ac.jp/2004/2/index.html>
- Emacs
 - <http://cns-guide.sfc.keio.ac.jp/2004/6/index.html>

10/1	基本知識の確認
10/8	Basic data structure
10/15	B+-tree insertion
10/22	B+-tree
10/29	Masstree
11/5	Masstree
11/12	Invited Talk
12/3	CCBench
12/10	CCBench
12/17	CCBench
12/24	Invited Talk
1/14	LineairDB
1/21	LineairDB

How to earn the credit?

- How to earn?
 - You will have some assignments.
 - Implement it.
 - Write a short report and have some feedback from TA.
- Deadline: next lecture
- Code
 - Place on CNS
- Report should include
 - Location of your code (directory)
 - Explanation of your code
 - The result of execution

Assignment 1: Hello

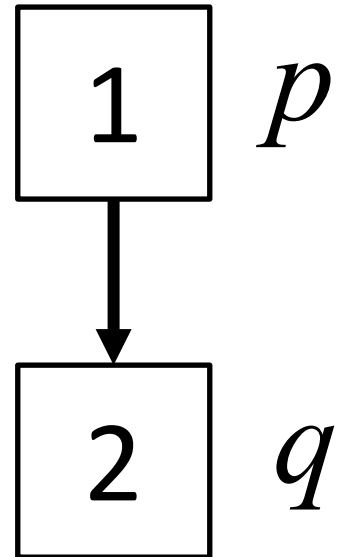
```
% ./hello  
Hello, SFC
```


List

```
/* list1.cc */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct list {
    struct list *next;
    int data;
};

int main(){
    struct list *p, *q;
    p = malloc(sizeof(struct list));
    p->data = 1;
    q = malloc(sizeof(struct list));
    q->data = 2;
    p->next = q;
    q->next = NULL;
    exit(0);
}
```



Assignment 2: List

- Create a list that has 100 elements. You do not need to implement (insert, delete)

```
% ./list
```

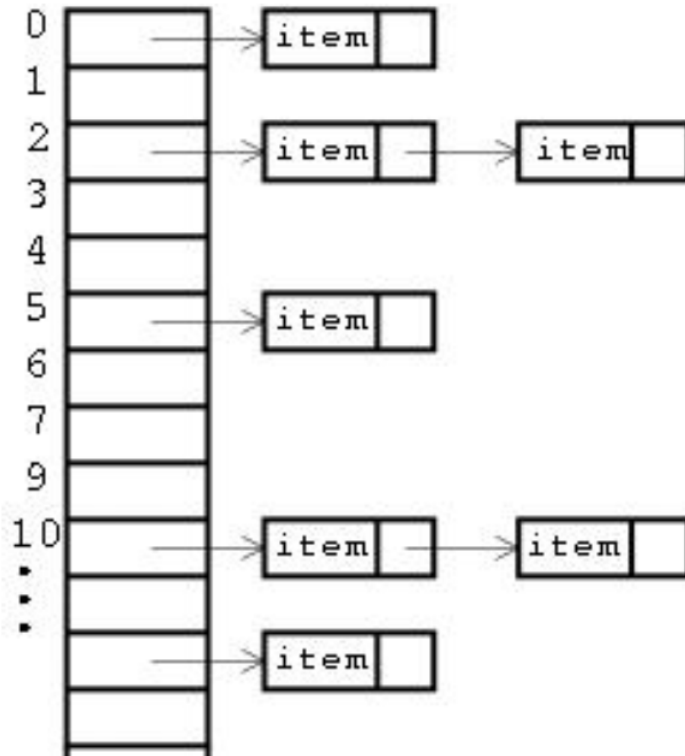
```
1 2 3 4 ...
```

Hash

- Hash?
 - a mixture of meat, potatoes, and vegetables cut into small pieces and baked or fried (Cambridge Dictionary)
- Hashing
 - Translate from key to address
 - The translation is conducted by hash function.
- Method
 - [Separate Chaining](#)
 - Open Addressing
- Ideal Complexity: $O(1)$

Separate Chaining

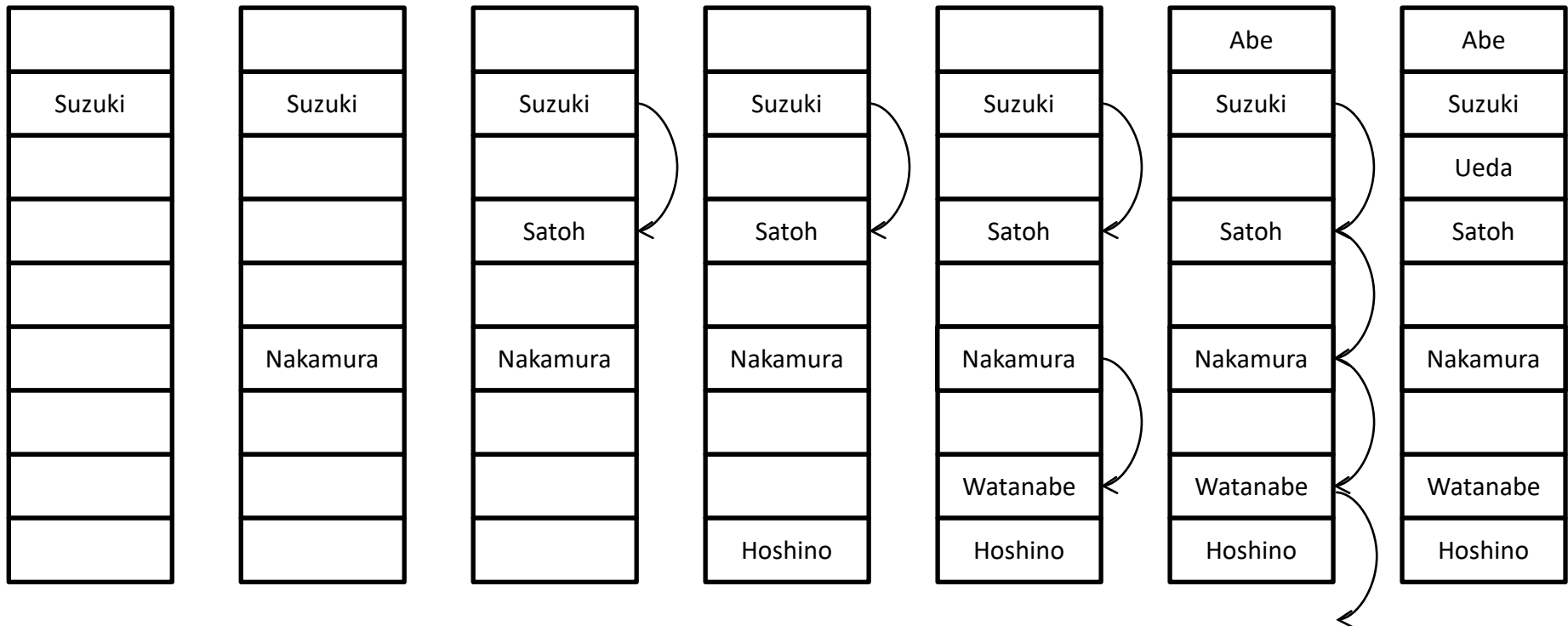
- Method
 - Linked list



- Quiz
 - Bucket Size ($M=10$)
 - 18, 90, 15, 85, 5,
70, 22, 40, 48, 96,
54, 62, 88, 60, 68,
77, 80, 7, 35, 50

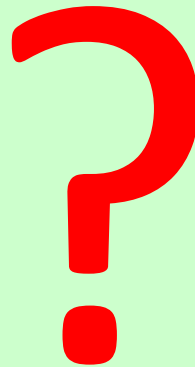
Open Addressing

- Bucket Size (M)=9, Delta=2
- Suzuki(19), Nakamura(14), Satoh(19), Hoshino(8), Watanabe(23), Abe(1), Ueda(21)



Quiz: Open Addressing

- Bucket Size (M)=9, Delta=2
- 15, 13, 14, 1, 7, 0, 9, 2, 3



Q. Open Addressing

- Show hash tables for the following inputs.
 - $M=13$, $d = 3$
 - 1. 18, 5, 12, 15, 7, 25, 31
 - 2. 6, 9, 19, 22, 16, 3

Assignment 3: Hash

- Implement hash with open addressing, if you have an interest.

% ./hash

1 2 3 4...

Key? 4

Found (Step = 1)

Report

- Code
 - Hash, binary-searchの性能比較
 - Assertを入れること
- Report
 - Place of code (github, private repository?)
 - Explanation of the code, briefly.
 - Execution result
 - hashがbinary searchに勝つ場合
 - hashがbinary searchに負ける場合
 - Less than 1 page (PDF)
- Deadline
 - By next lecture.

After submitting report,
you can ask TA to check
the code so that you can
have some feedback 😊