

東京大学  
情報理工学系研究科  
創造情報学専攻

Database

Shigeru Chiba

Web Programming Languages 02

Shigeru Chiba

[ 1 ]

## 感想より

- 包丁より箸と銀食器の対比の方が分かりやすかったのではないのでしょうか。  
日本には~~道具~~より技に頼る文化(美学?)は間違いなくある(った)と思います。
- 日本の道具をできるだけ少なく済ませる文化の代表例、考えてみたら風呂敷がありました。
- プログラミング言語は多すぎるためシングルランゲージの考えはすごくいいと思います。
- DSLでアプリケーション作ったら引き継ぎ相手が1人だけで大変なことになったのでシチュエーションによりけりだと思ひます

Shigeru Chiba

[ 2 ]

## 質問

- HTMLに埋め込まれたJavaScriptはクライアント側のブラウザで実行されていると考えていいのでしょうか。  
もしそうなら、HTMLを吐き出すJAVAの方はサーバーで実行されるしかない気がするのですが、両者でのサーバーでの計算コストなどはあまり気にしないものですか。

Yes

[ 3 ]

## 感想より

- PHPが使われる理由は  
デザインの変更がしやすいということもあると思いますが、  
早く世に出すということを考えると、小規模のサービスであればスピーディーに作れるからではないかと思います。

PHP is a DSL

[ 4 ]

## 感想より

ハッカーと画家

- (別にlispが好きなのではないですか……)  
(P

```
"Hello "
  (if (null? name) (STRONG "World") name)
  "!")
```

というようにプログラミング言語上のオブジェクトで表現するようにしておけば、複雑なロジックも実装できるし、デザインの変更も難しくない(むしろHTMLタグそのものを書き換えるより簡単?)ように思います。  
なぜそうならなかったのでしょうか(あるいはそうになっているのでしょうか?)

[ 5 ]

## 感想より

Ajax HTTP

HTTP

- HTTP or AJAX does not make sense for me!  
IMHO: AJAX or Relocation.
- Standard HTTP requests requires navigation to pages in order to change content, while AJAX uses such HTTP requests (GET/POST/..etc) under the same URL.
- JavaScriptから使えるHTTPではないプロトコルとしてはWebSocketが代表的

?

[ 6 ]

## 感想より

- Flash playerやAdobe acrobatは脆弱性の温床ですが、これはAdobeの細かいバグはチェックせず動くものを作って問題が発覚したら直すという方針のせいだとセキュリティ界隈で聞いたことがあります。真偽はわかりませんが。

[ 7 ]

## 感想より

- ブラウザで動く言語としてJavaScriptだけでなく近年WebAssemblyが注目されています。
  - あと asm.js? *binary data*

[ 8 ]

## 課題 1 by July 2

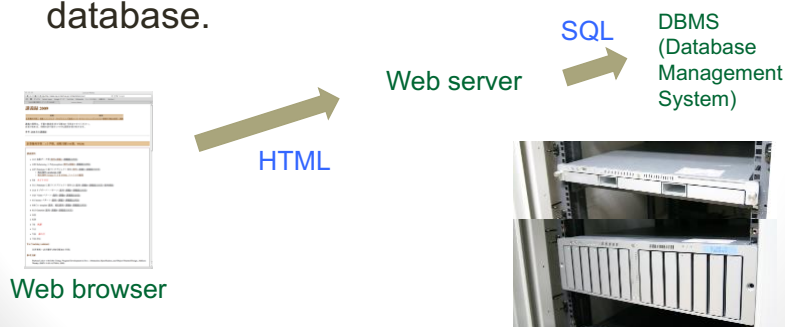
- Java Servlet のような printing 方式と、Java JSP, PHP, Ruby erb などの embedding 方式の利害得失を、具体的なコード例を示しながら述べよ。

第 4 週の講義内容を参考に、A4 用紙 1 ページ程度書くこと。提出は pdf ファイルか word ファイルとする。

[ 9 ]

## Relational Database *e*

- Everything important is stored in a database.
- IBM's DB2 in 1983, Oracle DB in 1977, ...
- A web application is just a front-end of database.



## Transaction (TX)

- A series of DB accesses
  - Commit
    - All the changes are written on the DB at once.
  - Rollback (or Abort)
    - All the changes are discarded without being written.
- Why needed?
  - Concurrency control and recovery

[ 11 ]

## Bank account

- Account a1 = 5000 yen.
- Transaction 1 (withdraw 1000yen)
  - k = read a1
  - write (k - 1000) to a1
- Transaction 2 (withdraw 2000yen)
  - j = read a1
  - write (j - 2000) to a1

[ 12 ]

## Concurrent execution

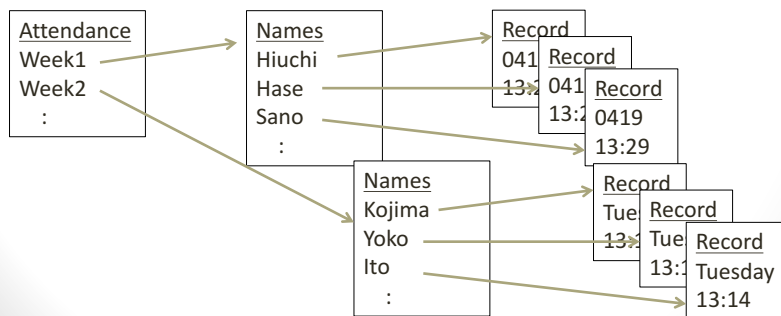
Transaction 1: k = read a1 write (k - 1000) to a1	Transaction 2: j = read a1 write (j - 2000) to a1
---	---

Transaction 1: k = read a1 write (k - 1000) to a1	Transaction 2: j = read a1 write (j - 2000) to a1
---	---

( 13 )

## How to store data

- Access speed is a critical issue.
- Hierarchical database
  - IBM's IMS in 1968



( 14 )

## CAP theorem for distributed systems

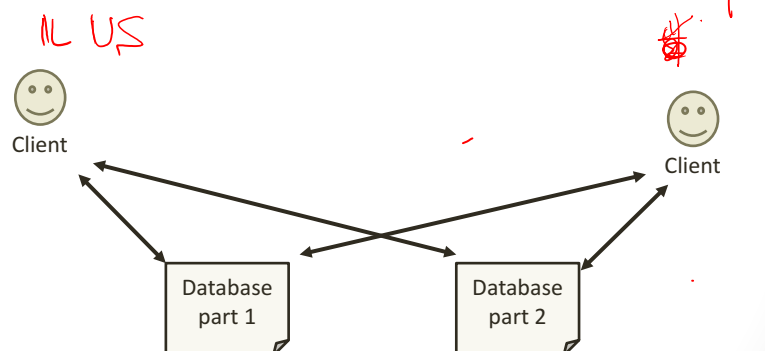
- Not possible to guarantee all the three:
- Consistency 一致性
  - all nodes see the same data at the same time.
- Availability 可用性
  - the data are always accessible.
- Partition-tolerance
  - the system works under a failure of sub systems

Shigeru  
Chiba

[ 15 ]

## Consistent & Available

- No partition-tolerance

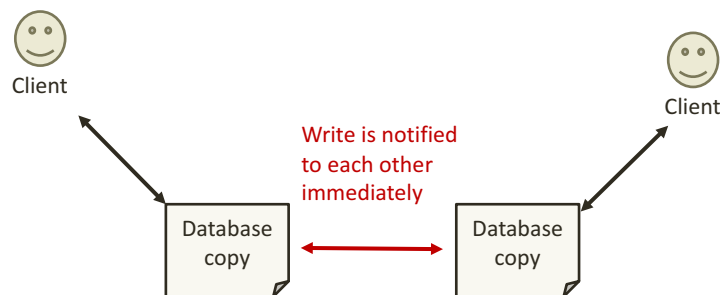
Shigeru  
Chiba

[ 16 ]



## Consistent & Partition-tolerant

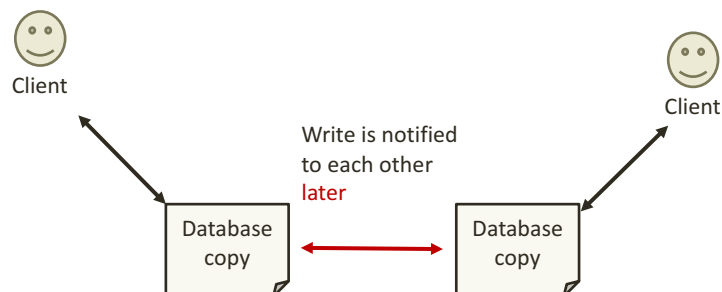
- No availability



[ 17 ]

## Available & Partition-tolerant

- No consistency



[ 18 ]

# NoSQL

1 / \*

- (Distributed) key-value store
  - Google's BigTable in 2004
  - How do you implement a query?

Key	week	name	time	passcode
1/Hiuchi	1	Hiuchi	13:27	0419
1/Hase	1	Hase	13:28	0419
1/Sano	1	Sano	13:29	0419
2/Kojima	2	Kojima	13:10	Tuesday

:

( 19 )

# Relational data model

- The model for relational databases
  - by E. F. Codd in 1969
- Based on the set theory.
  - independent of implementation
- Definition
  - $t \in R \subset D \times D \times D \times \dots \times D$
  - t: tuple, R: relation, D: domain

( 20 )

## Also represented by tables!

- Relation R is represented by a table.
  - row: tuple, column: attribute
- Table
  - no duplicated tuples in a table.
  - no order of tuples or attributes.
- Key
  - an attribute for uniquely identifying a tuple
  - a primary key

Shigeru  
Chiba

[ 21 ]

## Relational algebra

- Set operators
  - Union, difference, intersection, Cartesian product
- Other operators
  - Projection, selection, restriction, join, division

Shigeru  
Chiba

[ 22 ]

## Selection

- select tuples matching given criteria.
- in SQL
  - `select * from attendance where week=1;`

attendance

week	name	time	passcode
1	Hiuchi	13:27	0419
1	Hase	13:28	0419
1	Sano	13:29	0419
2	Kojima	13:10	Tuesday

:

{ 23 }

## (Inner) Join

- Combine two tables.
  - `select * from attendance, comments where attendance.ID=comments.ID;`

attendance

ID	week	name	time	passcode
10	1	Hiuchi	13:27	0419
11	1	Hase	13:28	0419
12	1	Sano	13:29	0419
13	2	Kojima	13:10	Tuesday

comments

ID	comment
11	Hello
12	Enjoyed!

{ 24 }

## (Inner) Join

- Combine two tables.
  - `select * from attendance, comments`  
`where attendance.ID=comments.ID;`

attendance

ID	week	name	time	passcode
10	1	Hiuchi	13:27	0419
11	1	Hase	13:28	0419
12	1	Sano	13:29	0419
13	2	Kojima	13:10	Tuesday

comments

ID	comment
11	Hello
12	Enjoyed!

joined

ID	week	name	time	passcode	comment
11	1	Hase	13:28	0419	Hello
12	1	Sano	13:29	0419	Enjoyed!

( 25 )

## (Outer) Join

- Combine two tables.
  - `select * from attendance outer join comments`  
`on attendance.ID=comments.ID;`

attendance

ID	week	name	time	passcode
10	1	Hiuchi	13:27	0419
11	1	Hase	13:28	0419
12	1	Sano	13:29	0419
13	2	Kojima	13:10	Tuesday

comments

ID	comment
11	Hello
12	Enjoyed!

joined

ID	week	name	time	passcode	comment
10	1	Hiuchi	13:27	0419	
11	1	Hase	13:28	0419	Hello
12	1	Sano	13:29	0419	Enjoyed!
13	2	Kojima	13:10	Tuesday	

( 26 )

## How to represent?

files << "Q1.pdf"

- Variable-length columns
- Use join!

ID	week	name	time	file	file	file
10	1	Hiuchi	13:27	Q1.pdf	Q2.pdf	
11	1	Hase	13:28	Q1.pdf		
12	1	Sano	13:29	Q1.pdf	Q2.pdf	Q3.pdf
13	2	Kojima	13:10	Q1.pdf		

[ 27 ]

## How to represent?

- Variable-length columns
- Use join!

submission

ID	week	name	time
10	1	Hiuchi	13:27
11	1	Hase	13:28
12	1	Sano	13:29
13	2	Kojima	13:10

files

ID	submit_ID	file
1	10	Q1.pdf
2	10	Q2.pdf
3	11	Q1.pdf
4	12	Q1.pdf
5	12	Q2.pdf
6	12	Q3.pdf
7	13	Q1.pdf

variable

[ 28 ]

## How to represent?

- **select** \* **from** submission, files  
  **where** submission.ID=files.submit\_ID;

ID	week	name	time	files.ID	file
10	1	Hiuchi	13:27	1	Q1.pdf
10	1	Hiuch	13:27	2	Q2.pdf
11	1	Hase	13:28	3	Q1.pdf
12	1	Sano	13:29	4	Q1.pdf
12	1	Sano	13:29	5	Q2.pdf
12	1	Sano	13:29	6	Q3.pdf
13	2	Kojima	13:10	7	Q1.pdf