Gagnasafnsfræði

Páll Melsted

18. nóv

JSON

JavaScript Object Notation (JSON)

- Staðall til að skrifa niður hluti (e. object) á mannamáli
- Notað til að skiptast á gögnum og til að geyma hálfformuð gögn
- Upphaflega var JSON byggt beint ofan á JavaScript

Dæmi

```
{ "Books":
  [
    { "ISBN":"ISBN-0-13-713526-2",
      "Price":85,
      "Edition":3,
      "Title":"A First Course in Database Systems",
      "Authors":[ {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
    ,
    { "ISBN":"ISBN-0-13-815504-6",
      "Price":100,
      "Remark":"Buy this book bundled with 'A First Course' - a great deal!",
      "Title":"Database Systems:The Complete Book",
      "Authors":[ {"First_Name":"Hector", "Last_Name":"Garcia-Molina"},
                  {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
  ],
  "Magazines":
  [
```

```
    { "Title":"National Geographic",
      "Month":"January",
      "Year":2009 }
    ,
    { "Title":"Newsweek",
      "Month":"February",
      "Year":2009 }
  ]
}
```

JSON skilgreining

Endurkvæm skilgreining

- Grunngildi: tölur, strengir, boolean, null
- Hlutir {}: mengi af label-value pörum
- Fylki []: listar af gildum

Gildin geta verið af hvaða tagi sem er, label verða að vera strengir.

JSON

Hlutverk JSON

JSON er ekki lengur bundið við Javascript. Fjöldi af forritasöfnum er til fyrir flest forritunarmál til að lesa og meðhöndla JSON.

Miðað við XML er JSON

- læsilegra, þægilegra til að skrifa beint t.d. config skrár
- passar betur við hlutbundna forritunarhugsun
- óhentugra fyrir stöðluð gögn, minna um schema lausnir
- hefur ekki jafngott fyrirspurnarmál eins og XML

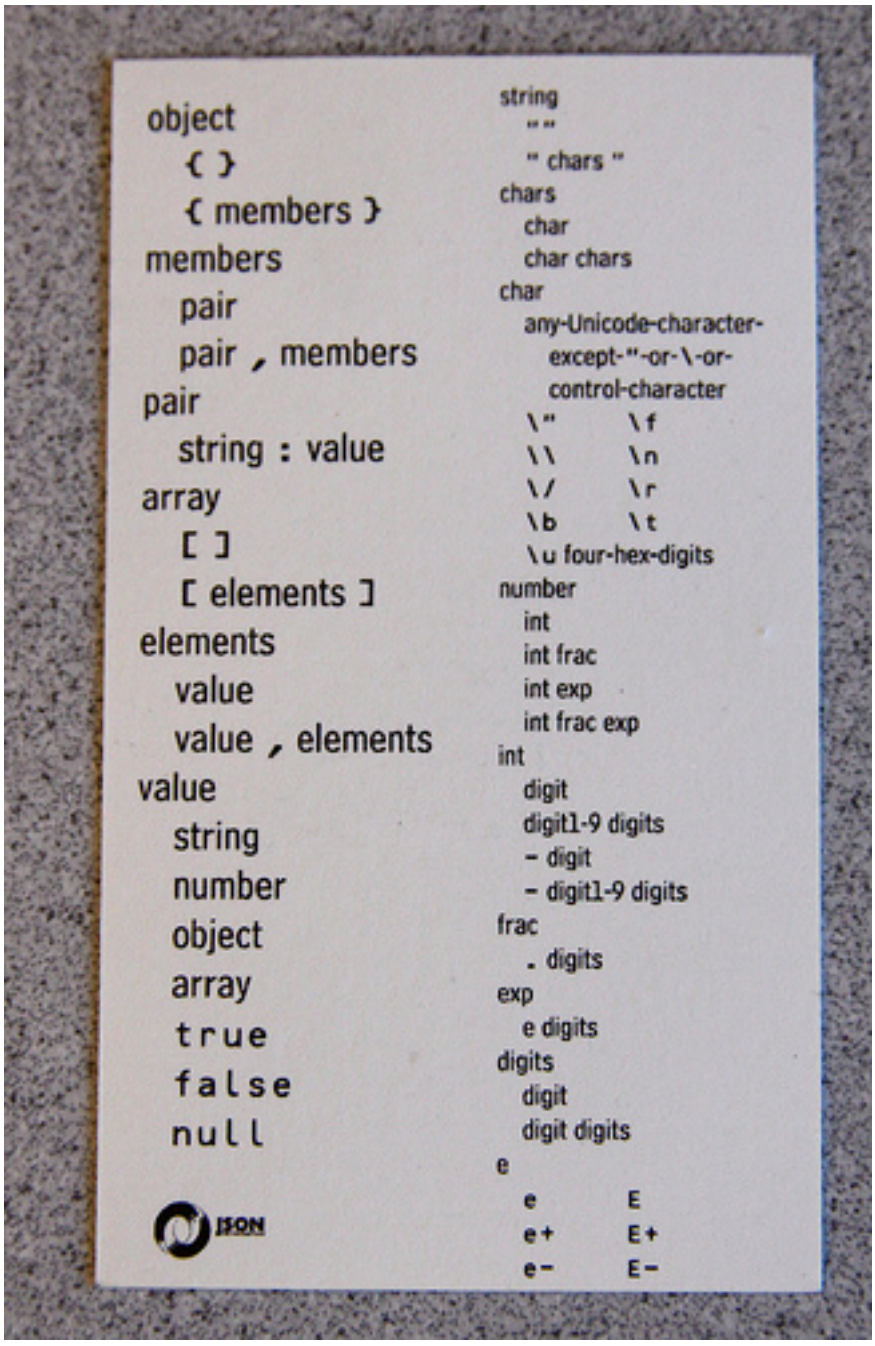
object
{ }
{ members }
members
pair
pair , members
pair
string : value
array
[ ]
[elements]
elements
value
value , elements
value
string
number
object
array
true
false
null
JSON
string
" "
" chars "
chars
char
char chars
char
any-Unicode-character-
except-"-or-\-or-
control-character
\" \f
\\ \n
\/ \r
\b \t
\u four-hex-digits
number
int
int frac
int exp
int frac exp
int
digit
digit1-9 digits
- digit
- digit1-9 digits
frac
. digits
exp
e digits
digits
digit
digit digits
e
e E
e+ E+
e- E-

Figure 1:

JSON í Python

Pakkinn `json` í python er notaður til að skrifa og lesa json.

- `json.load / loads` les JSON skrá / streng og skilar hlut
- `json.dump / dumps` skrifar JSON skrá eða skilar streng
- Strengir, tölur og boolean eru túlkuð beint, `null` verður að `None`
- Fylki verða `list`
- Hlutir verða að `dict`

MapReduce

MapReduce er forritunarumhverfi sem vinnur beint ofan á HDFS. Enginn gagnagrunnur, öll gögn eru geymd í skrám.

Útreikningar eru framkvæmdir í þremur fösum

1. map: breytir gögnunum
2. group: hópar saman lík gögn
3. reduce: samantekt innan hópa

Notandi þarf bara að útfæra tvö föll.

MapReduce

MapReduce umhverfið sér um að dreifa útreikningum á vélar, fylgjast með framgangi og jafna sig á áföllum

---

map

map skrefið tekur gögn úr skrá, yfirleitt ein lína og býr til lista af `(key,value)` gildum.

Endurtekningar eru leyfðar og gildin á `key` þurfa ekki að vera ólík.

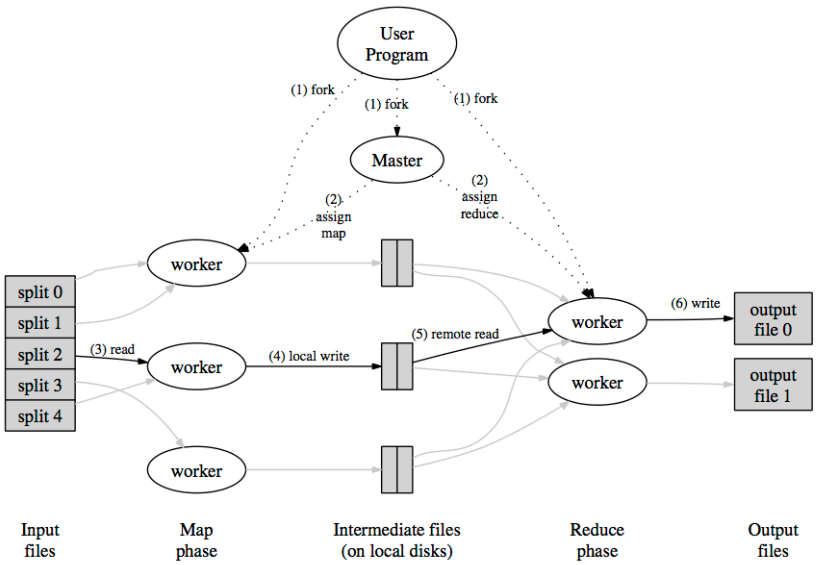

Figure 1: Execution overview

Figure 2:

group

group skrefið tekur öll `(key,value)` gildin og flokkar eftir gildinu á `key`

Öll `value` gildi fyrir sama `key` eru sett í lista.

Notandi þarf ekki að gera þetta, MapReduce umhverfið sér um þetta skref.

---

reduce

reduce skrefið leysir hvert vandamál fyrir sig og skilar út niðurstöðunni

- Inntakið er `(key,list)` þar sem `list` er listi af `value` gildum
- Úttakið er reiknað út frá listanum

Úttak fyrir öll `key` gildi er sett í eina skrá.

SQL í MapReduce?

Lausnin er "einföld", við umritum SQL í venslaalgebru og sjáum hvernig helstu virkjar eru útfærðir.

Við gerum ráð fyrir að gagnagrunnurinn hafi þekkt skema og að inntakið fyrir töflu $A(a_1, \ldots, a_n)$ sé táknað með línu á forminu

```
A a1 a2 ... an
```

Við munum síðan sleppa nafninu A í úrvinnslu ef við erum aðeins að vinna með eina töflu.

Fyrir hverja aðgerð þurfum við að skilgreina map og reduce fall.

Val $\sigma_C(R)$

Þar sem C verður að vinna með dálkana í R er hægt að athuga skilyrðið á hverri línu fyrir sig.

map:

- n-din t varpast í (t, t), ef $C(t)$ er satt.

reduce:

- ekkert! þetta er þá samsemdarvörpunin

Vörpun $(\pi_L(R))$

map:

- n-din t varpast í (t', t') þar sem t' er t með dálkana úr L

reduce:

- Inntakið fyrir reduce verður á forminu $(t', [t', \ldots, t'])$ og breytir því í (t', t')

Hópun

Gerum ráð fyrir að við viljum finna fyrir $R(A, B, C)$

```
SELECT A,MAX(B)
FROM R
GROUP BY A
```

map:

- Inntakið (a, b, c) varpast yfir í (a, b)

group skrefið í mapreduce sér til þess að öll a gildin hópast saman

reduce:

- Inntakið er $(a, [b_1, \ldots, b_n])$ sem varpast í $(a, \max(b_1, \ldots, b_n))$

Sammengi $R \cup S$

map:

- $(R, t) \to (t, t)$ og $(S, t) \to (t, t)$ sama vörpun fyrir báðar töflur

reduce:

- Inntakið verður á forminu $(t, [t, t])$ eða $(t, [t])$ og skilar (t, t)

Mismunur $R \setminus S$

map:

- Vörpum inntakinu $(R, t) \to (t, R)$ og $(S, t) \to (t, S)$

reduce:

- Vörpum $(t, [R]) \to (t, t)$, allt annað fer í ruslið

Tenging $R \bowtie S$

Gerum ráð fyrir að $R(A, B)$ og $S(B, C)$ og reiknum $R \bowtie S$

map:

- Vörpum $(R, a, b) \to (b, (a, R))$ og $(S, b, c) \to (b, (c, S))$

reduce:

- Við skilum gildi aðeins þegar b kom fyrir bæði í R og S. Vörpunin er því $(b, [(a, R), (c, S)]) \to (a, b, c)$. Ef það eru margar n-dir í R eða S með sama b gildi þá búum við til allar samsetningar.

Hadoop

Flestir dreifðir gagnagrunnar byggja ofan á Apache Hadoop

- Apache Hive - notaður fyrir vöruhús gagna, HiveQL svipað og SQL
- Apache Pig - einfaldara forritunarmál til að búa til map-reduce forrit. Hefur ýmsa SQL eiginleika
- Apache HBase - NoSQL "BigTable" gagnagrunnur
- Apache Cassandra - NoSQL "BigTable" gagnagrunnur

BigTable gagnagrunnar nota tiltölulega laust skema, það er hægt að bæta við dálkum eftir á og eðlilegt að hafa fjölda dálka. BigTable er frá Google en engin útfærsla er til.

PageRank

Pagerank er reiknað út frá fylkinu P sem er $n \times n$ fylki fyrir n vefsíður. Hver dálkur samsvarar síðu og $P_{i,j} = 1/d_j$ ef síða j hefur hlekk á síðu i og fjöldi hlekkja á síðu j er d_j . Pagerank π er reiknað sem lausnin á jöfnunni

$\pi = \beta \cdot \pi P + (1 - \beta)\frac{1}{n}$

Einfaldasta lausnin er að nota ítrun til að reikna út π. Við byrjum með $\pi = (\frac{1}{n}, \ldots, \frac{1}{n})$ og beytum jöfnunni aftur og aftur. Dæmigerð gildi á β er ca 0.85.

Pagerank og MR

Við gerum ráð fyrir að inntakið sé raðað þannig að hver lína innihaldi síðunafn og hlekki á allar síður, þ.e. dálkurinn í P fylkinu, P_j. Einnig gerum við ráð fyrir að núverandi ágiskun fyrir PageRank fylgi með inntakinu fyrir síðuna.

map:

- Inntakið er (j, π_j, P_j) og úttakið er n-dir $(i, \pi_j \cdot P_{i,j})$ fyrir alla hlekki i

reduce:

- Inntakið er $(i, [\pi_j \cdot P_{i,j}, \ldots]])$ úttakið verður $(i, \beta \sum_j \pi_j \cdot P_{i,j} + (1 - \beta)/n)$