MTA SZTAKI
Magyar Tudományos Akadémia
Számítástechnikai és Automatizálási Kutatóintézet

Longneck Data Integration
open source data quality eszköz

MTA SZTAKI

Üzleti Intelligencia és Adattárházak Csoport

Big Data Üzleti Intelligencia Csoport

Sidló Csaba

sidlo@sztaki.mta.hu

http://longneck.sztaki.hu
http://dms.sztaki.hu
http://bigdatabi.sztaki.hu

2013. november 6.

MTA SZTAKI Informatika Kutatólabor

- teljes innovációs lánc, **kutatástól alkalmazásokig**
  - adatbányászat, gépi tanulás, keresőtechnológiák, üzleti intelligencia, adattárházak, szociális hálózatok, bioinformatika
  - **„Big Data”**:
    - Web, közösségi média elemzés és keresés
    - smart city, mobilitás
    - szenzoradatok (pl. szélerőmű), log-adatok
    - dedikált csoportok: „Lendület – Big Data”, „Big Data Üzleti Intelligencia” (partner: SZTAKI EMI)

vodafone

AEGON

RICHTER GEDEON

[origo]

Magyar Telekom

StratoSphere
Above the Clouds

LiWA
Living Web Archives

columba
GLOBAL SYSTEMS

SEVENTH FRAMEWORK PROGRAMME

LAWA

BAE SYSTEMS

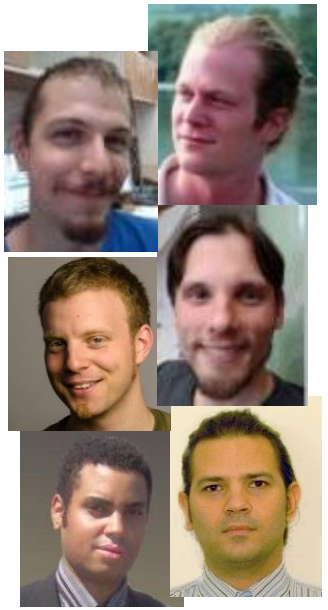
indra

denodo

FINMECCANICA

Longneck történet

- eredeti **feladatok**:
  - adattárház: ügyféladatok
  - Web szerver + általános IT infrastruktúra log analitika
  - szenzor-adat feldolgozás
- **megoldások**:
  - egyedi scriptek 2005
  - „Giraffe ETL” 2009
    - folyamathoz generált kód
    - open source: 2010
  - új általános ETL eszköz
  - Longneck: Java data quality tool
    - open source: 2013



Lukács Gábor
Molnár Peti
Molnár András
Neumark Peti
Németh Tibi
…

http://longneck.sztaki.hu
https://github.com/MTA-SZTAKI/longneck-*

„Yet Another ETL Tool”?

"**Don't build** your own ETL framework"
"STAY AWAY from scripting languages"

"If any of the existing ETL frameworks are not powerful enough, **build your own!**"

„BI that avoids ETL processes entirely”

„ETL is a methodology, not a tool”

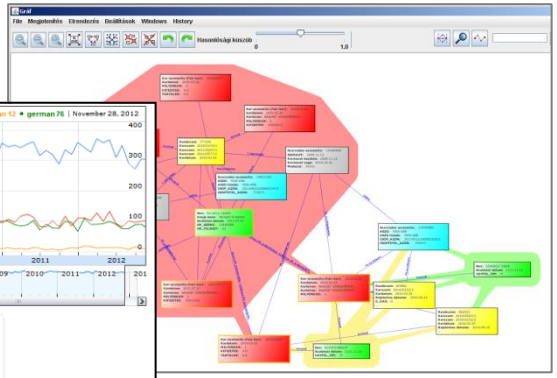
- Saját eszköz?
  - sok projektünkhöz kellett
  - most: a **Longneck-et választanám**
  - régen (jobbak persze ma már):
    - Longneck nem volt, viszont
    - Pentaho Kettle:
      - lassú (teszt: file egy az egyben másolása)
    - kommerciális eszközök (pl. IBM Quality Stage, SAP, SAS, …)
      - drága, nem elég rugalmas / nem elérhető egymagában
    - nem lehetett jó minőségben transzformációkat implementálni!
- Nyílt forráskódú, ingyen?
  - keretrendszer vs. szabálykészlet értéke
    - a valódi érték a leírt iparági, üzleti tudásban van!
  - kívülről is fejlődhet

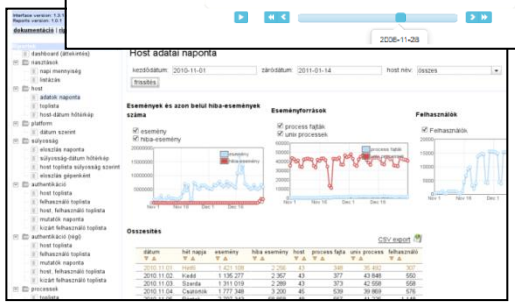
ETL / Data Integration / Quality Tool / …

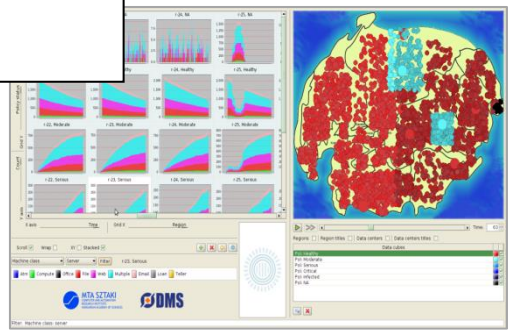
- heterogén, átfedő definíciók, nehezen általánosítható feladatok
- Gartner „Data Quality Tools Magic Quadrant” jobb felső:
  - Informatica, SAS / DataFlux, Trillium Software, SAP, IBM
  - Open Source eszközök?
    - pl: Talend – érdemes reklamálni, bekerültek riportokba

kapcsolódó eszközeink:

- **azonosságfeloldás** (entity resolution, matching)
- ETL, job management
- riportozás, analitika, vizualizáció







Adatminőség javítás: a feladat

- „Data Quality Tool” (Gartner):
  - **Parsing and standardization**
  - **Generalized cleansing**
  - **Matching**
  - Profiling
  - **Monitoring**
  - **Enrichment**

pl.: email cím

```
nincs@freemail.hu   → üres
maik@szarvnet.huhu → maik@szarvnet.hu | maik |
                      szarvnet | hu | Hungary
wwwcsomosmp@hu      → hibajelzés: „invalid e-mail
                      address(es).”
cokos@indamail.hu ; cokos@freemail.hu →
                      két valid email címre bontás
```

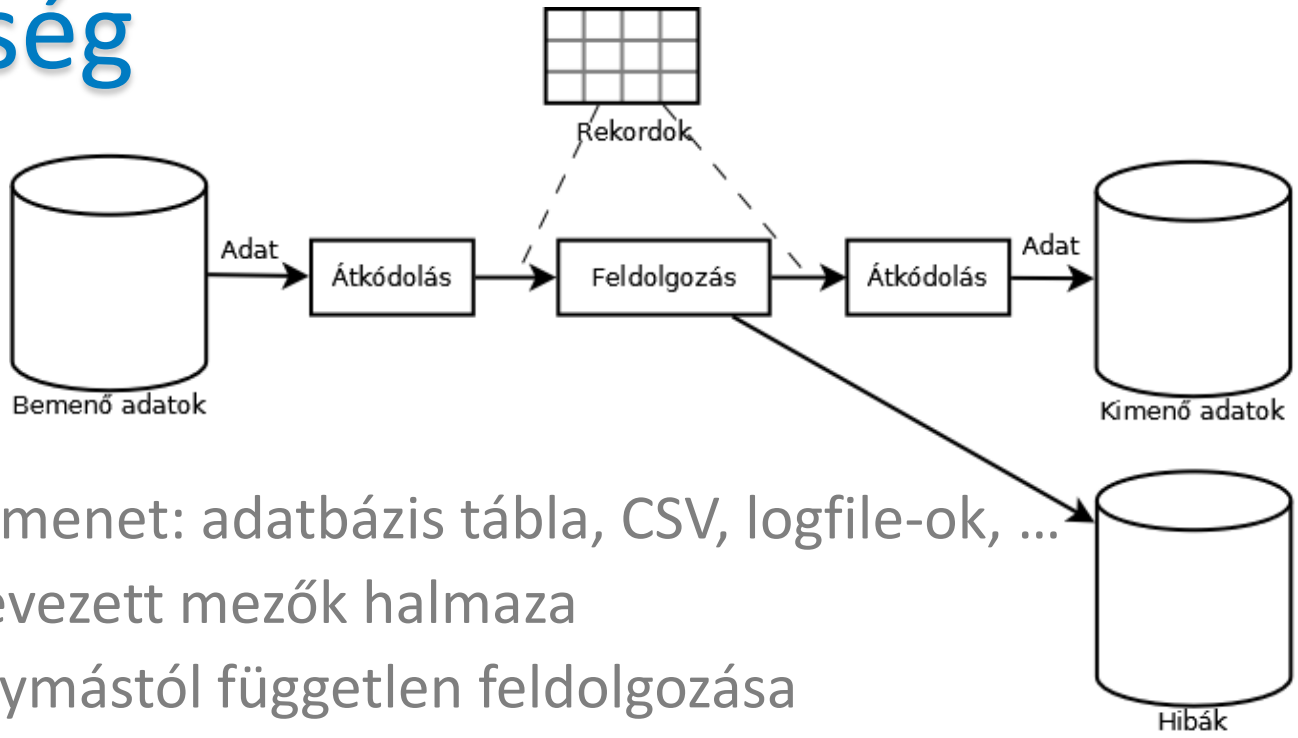
- elvárások adatminőség javító megoldással szemben:
  - gyors!
  - pontos!
  - stabil, robusztus!
  - könnyen fejleszthető!
  - újrafelhasználható!
  - …
  - olcsó, rövid projektek!
  - sok adatot képes kezelni!
  - olcsó, megbízható support!
  - könnyű módosítani!
  - „jól” kezeli a hibákat!
  - …

Data vs. software quality

”Pilot erejéig mind nagyon szépen működik”

- **transzformáció-leírások** vs. **keretrendszer** minősége: mindkettő feltétele a jó adatminőség előállításának
- jó minőségű megoldás (mint software mérnök):
  - megbízhatóság, biztonság, adatbiztonság,
  - mérhetőség, nyomonkövethetőség,
  - karbantarthatóság, adaptálhatóság, modularitás, hordozhatóság
  - dokumentáltság, könnyen elérhető tudás (szakértők ára),
  - tesztelhetőség (deploy után nem másnap látom a hibát)
  - hatékonyság, skálázhatóság, gyorsaság,
  - …
- Tudják ezeket a meglévő eszközök? Mennyiért?

Egyszerűség



- kimenet, bemenet: adatbázis tábla, CSV, logfile-ok, …
- **rekord**: elnevezett mezők halmaza
- rekordok egymástól független feldolgozása
  - **adatfolyam** szemlélet
- **típustalan**, minden érték String
  - pl: Java JDBC, DB → program → DB: 8-10 típusváltás
- **entitások**: elvárások különválasztása, leírása
  - ismert szabványok alapján
- **blokkok**: transzformációs szabályok
  - egymásba ágyazható, moduláris, specializálható, verziózható

terv:

XML, JSON, félig strukturált adatok jobb támogatása (bővítmény)

Repository

- pl.: zárt formátumú repository horror sztorik
- Longneck: XML

```
<process xmlns="urn:hu.sztaki.ilab.longneck:1.0">
  <source>
    <weblog-file-source name="cli" />
  </source>
  <target>
    <csv-target target="out.csv"/>
  </target>
  <blocks>
    <block-ref id="weblogparser:parse"/>
    <if>
      <is-null apply-to="virtualServer"/>
      <then>
        <copy apply-to="virtualServer"
                  from="requestUrlHost"/>
      </then>
    </if>
    <check summary="virtualServer must not be null.">
      <not-null apply-to="virtualServer"/>
    </check>
  </blocks>
</process>
```

- verziókövetés: saját + GIT
- kollaboratív munka támogatás
- újrahasznosítás, modularitás:
  - egymásba ágyazható
  - specializáció
  - bővíthető
- téma-orientált
- nyílt szabványok, coding convensions

Repository 2.

- kicsit bőbeszédű; talán jobb lenne, így utólag, ma már: domain specific language, pl. Groovy-val

```
process {
  source {
    weblog-file-source "cli"
  }
  target {
    csv-target "out.csv"
  }
  blocks {
    block-ref "weblogparser:parse”
    if (is-null „virtualServer") {
      copy from: "requestUrlHost" to: "virtualServer"
    }
    check summary: "virtualServer must not be null." {
      not-null "virtualServer"
    }
  }
}
```

- **a Longneck könnyen bővíthető!**
- pl:
  - lookup dictionary
  - DNS resolver
  - GeoIP
  - BDB perzisztencia
  - log-parser
  - webes látogató azonosítás
  - GeoLocation
  - …

- implementáljuk ezt az alternatívát is

Műveletek

```
<source>
  <database-source connection-name="client-db" />
    <query>
      select *
      from   p1_client c1
               join p2_client c2 on (c1.id=c2.id)
    </query>
 </database-source>
</source>
```

- join, rendezés, group by: SQL adatbázis, Hadoop stb.
  - ezekben jók!
  - fölösleges még egy réteget ráhúzni, SQL pl. szép, deklaratív, jól működik
- kontroll struktúrák: szekvencia, beágyazás, if-then, case, …
- regexp és string alapú felbontások, cserék, tesztek, …
- szótár alapú behelyettesítések, ellenőrzések, …

Meglévő transzformáció-készletek

- név, cégnév, adószám, TB szám, nem, születési dátum, …
- postai cím, geolocation, telefonszám, emailcím, …
- webanalitikai események, http request, URL, user agent, IP cím, …
- applikációs log események, attribútumok
- szenzor-adatok (pl. szélerőművek, location - smart city)

Robusztusság, hibakezelés

„do not let your ETL layer to corrupt or push wrong data into destination data layer”

- Longneck: egy **túlélőművész**! – mindig lefutnak a processek
  - adathibák: **lokális hatások**
  - beállítható szintű részletes logolás → felderíthetőség, tesztelhetőség

name	tax_marking	address	phone
Sidló Csaba Dr	11111111	1111 Bp Kende 13-17 XA	(20)39323872

kimenet:

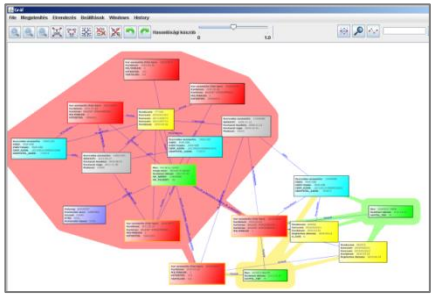
Dr. Sidló Csaba		1111, Budapest, Kende, utca, 13-17	

hiba kimenet:

postal-address	1	Address error.	1111 Bp Kende 13-17 XA
house-number-unit	2	Address err.: invalid house number unit.	XA
phone	1	Phone num. err.	(20)39323872
phone-number	2	Phone num. err.: invalid number.	39323872

Tesztelhetőség, mérhetőség

- input, output mérete, hibák száma, futásidők, …
- egyszerű és bevált tesztelő módszerek
  - blokkok tesztelése: egyedi adatok könnyű megadása – még nem parancssori, de lesz

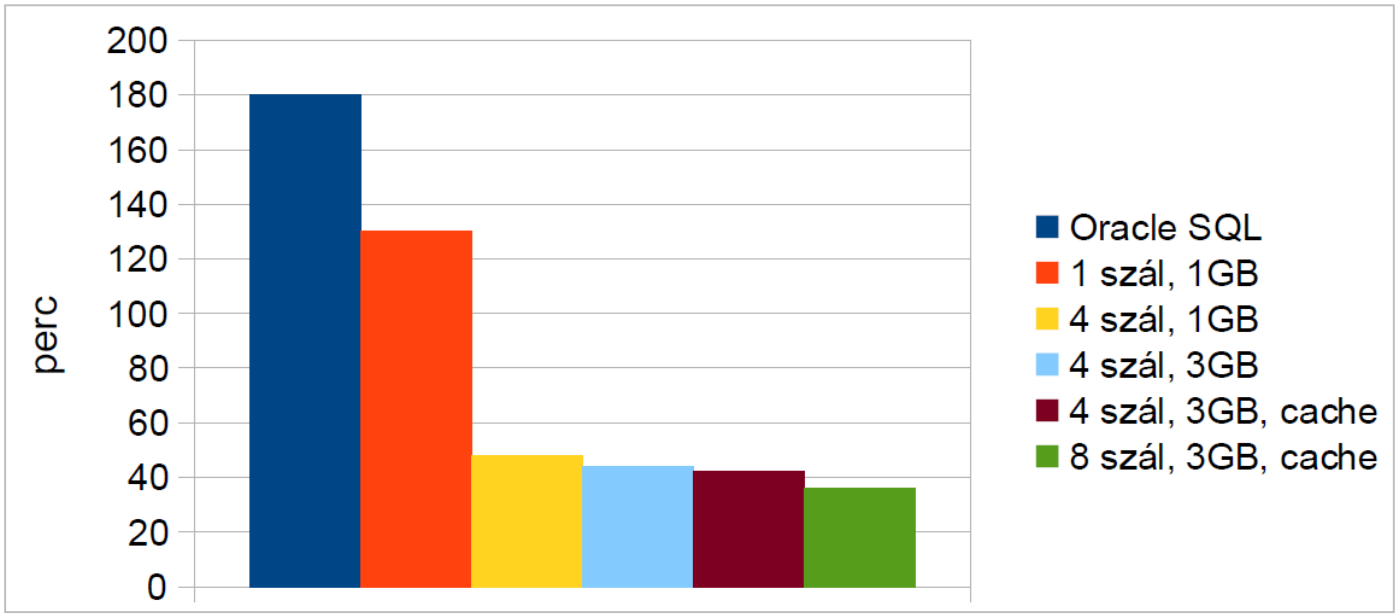


GUI

- Nincs. Kell?
- jó lenne; üzlet számára egyszerűen érthető leírás kell
- viszont az XML repo nem kihagyható:
  - gyorsan szerkeszthető, precíz, könnyű deploy-olni stb.
  - pl: GUI-n folyamat 100 szabályának apró módosítása

Skálázhatóság és Big Data

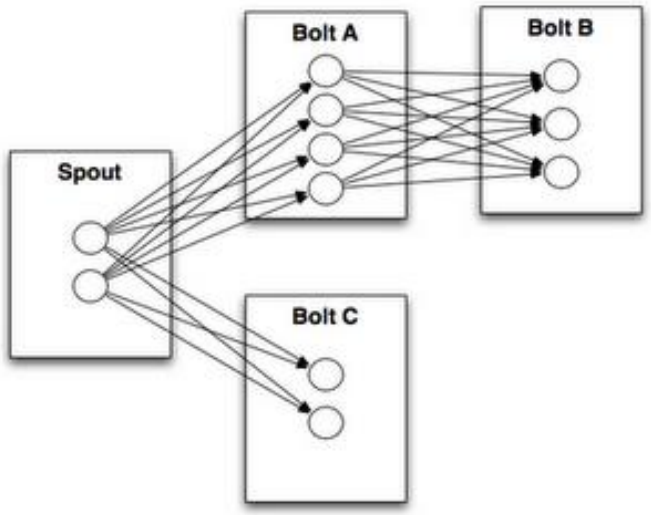
- a Java futtató **szálakra bomlik**
- pl: ~12M alkalmazás log sor, authentikációs események szűrése és transzformációi, gyenge gép, sok regexp illesztés:

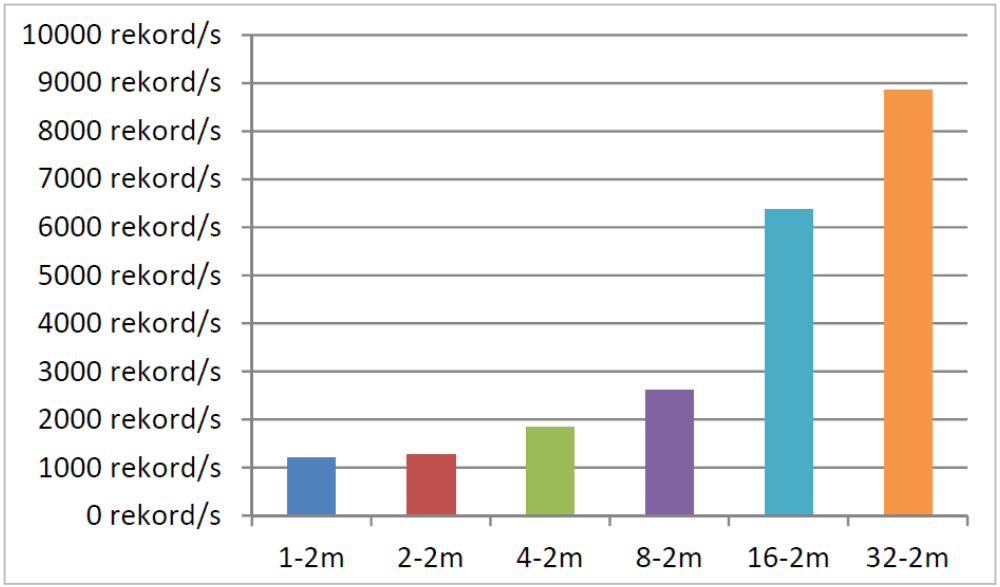


- szépen skálázódik
  - 8 szálnál már I/O korlát, memóriát igazából nem fogyaszt
- de még nem „Big Data”

Longneck-Storm

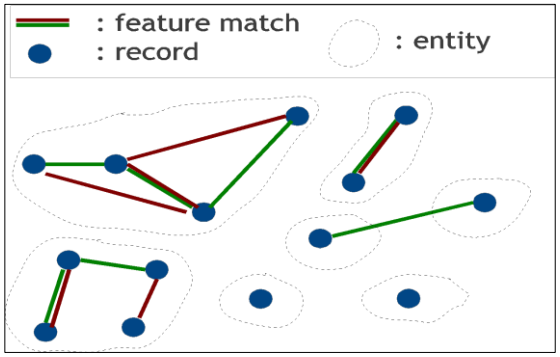
- **Hadoop**: kötegelt feldolgozás
  - megoldható, most nincs futtató
- **Storm**: adatfolyam feldolgozás valós időben
  - automatikus feladat elosztás
  - garantált adatfeldolgozás
- **Longneck-Storm:**
  - egyszerű futtató Storm klaszterre
- példa: Webszerver log
  - 32 gyengébb gépen
  - input: memóriából
  - **közel lineáris!**



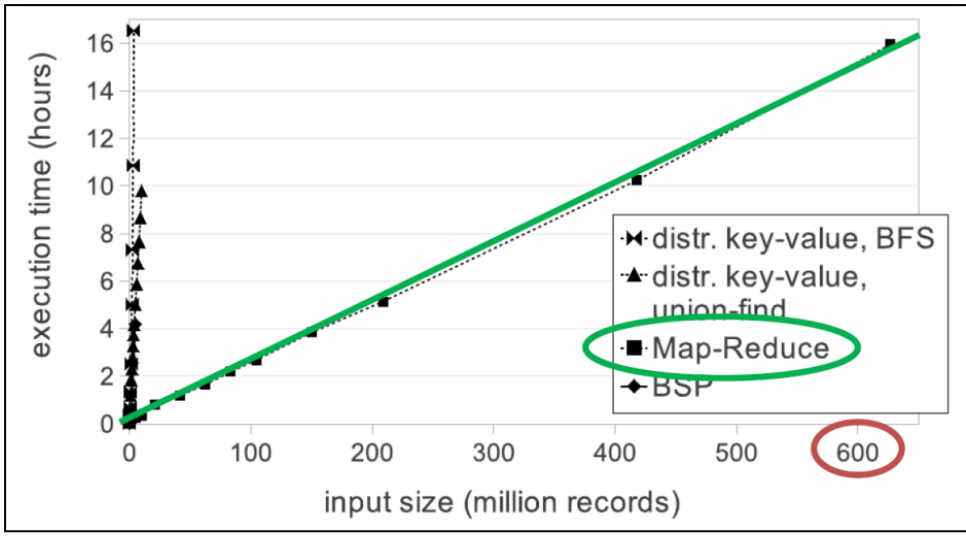


Kapcsolódó megoldásunk: azonosságfeloldás

név	e-mail	ID
Kovács Mária	marcsi@mail-1.com	50071
Nagy Istvánné	nagyne.marcsi@mail-2.hu	50071
Nagy Istvánné K. Mária	nagyne.marcsi@mail-2.hu	79216
Kovács M.	marcsi@mail-1.com	34302



- „Hány ügyfelünk van igazából?”
- duplikátumok → rekord-csoportok
- **Longneck + azonosságfeloldás**: nagyon erős kombináció
- más eszközök: ~10M rekordig max.
- Hadoop: kötegelt, Strom: valós idejű implementációk



VLDB konf., 2011,
Quality in Databases Workshop

Sidló Csaba

sidlo@sztaki.mta.hu

http://longneck.sztaki.hu -- próbálj ki!

http://dms.sztaki.hu

http://bigdatabi.sztaki.hu