INFOBRIGHT

New Features in the Infobright Community Edition (ICE)

Dominik Ślęzak

Outline

- Introduction
- Case Studies
- ICE/IEE RDBMS
- DomainExpert
- Rough Query

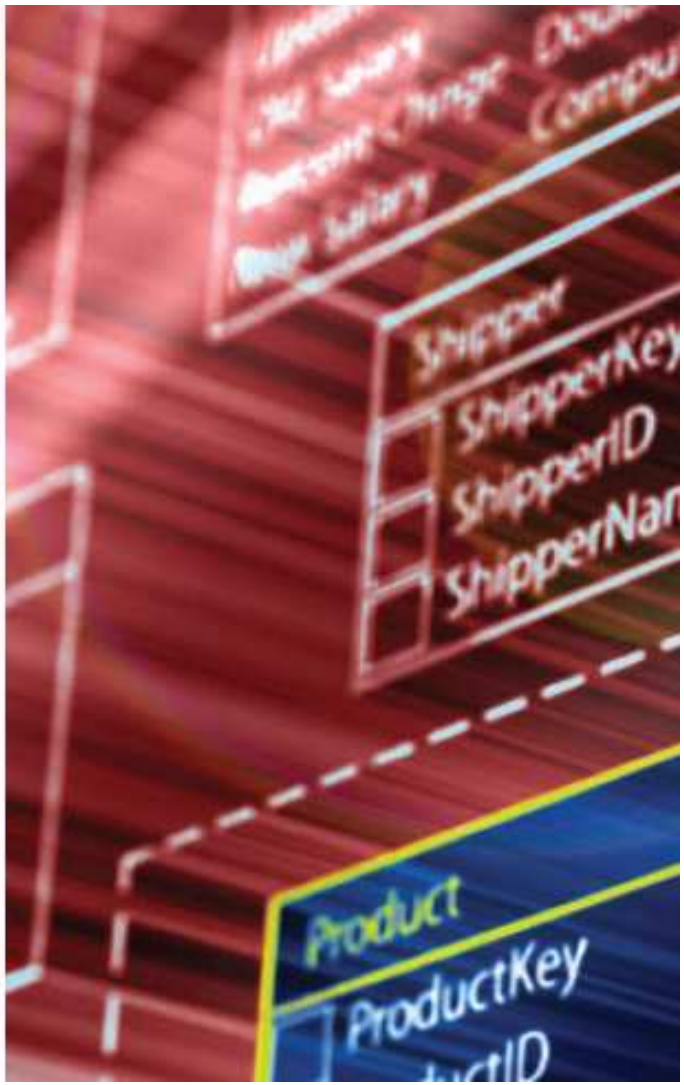
Communications ACM 08/2011 Vol.54 No.8

DOI:10.1145/1978542.1978562

BI technologies are essential to running today's businesses and this technology is going through sea changes.

BY SURAJIT CHAUDHURI, UMESHWAR DAYAL, AND VIVEK NARASAYYA

An Overview of Business Intelligence Technology



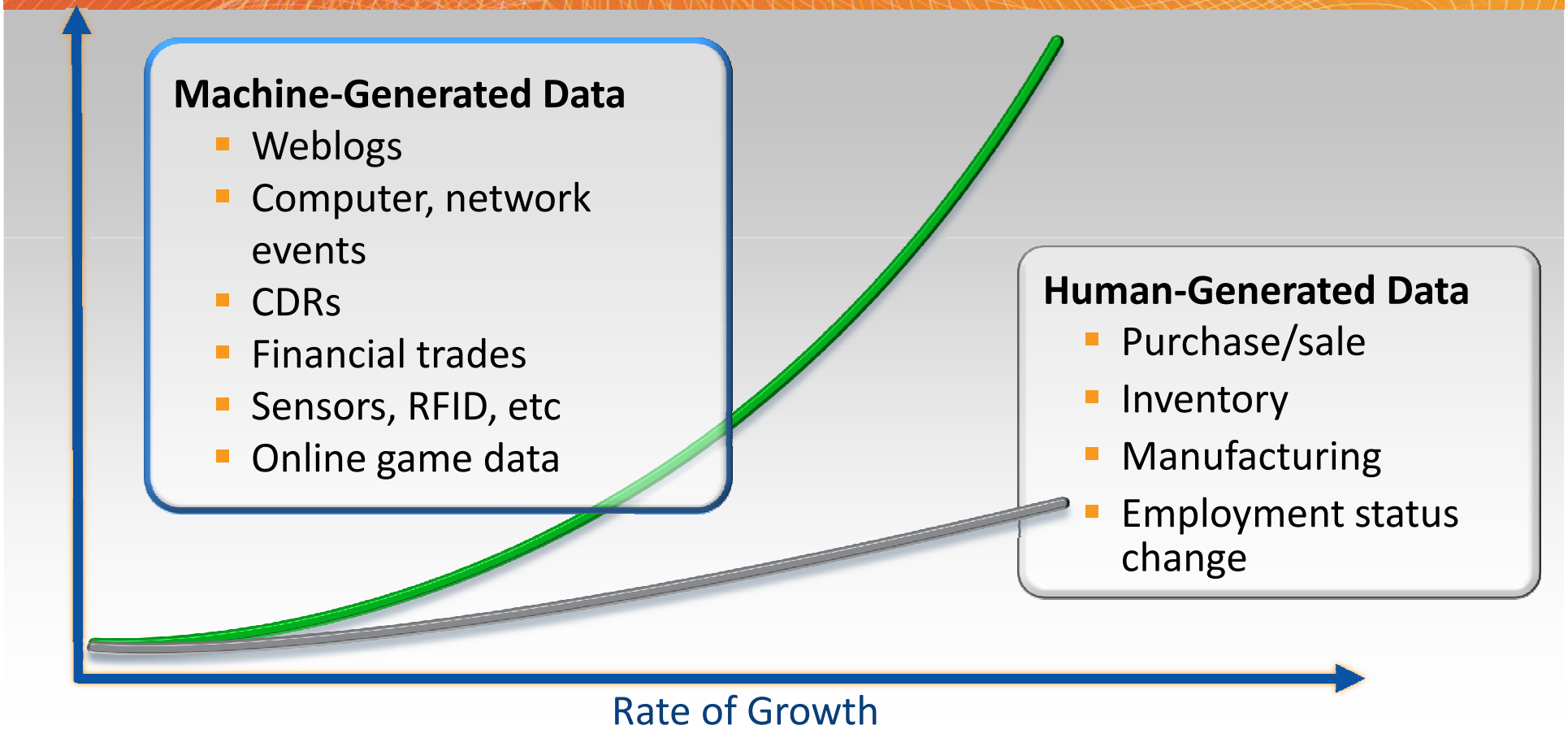
INFOBRIGHT

Communications ACM 08/2011 Vol.54 No.8

- The cost of data acquisition and data storage has declined significantly. This has increased the appetite of businesses to acquire very large volumes in order to extract as much competitive advantage from it as possible.
- The need to shorten the time lag between data acquisition and decision making is spurring innovations in business intelligence technologies.

The Machine-Generated Data Challenge

Fastest-Growing Category of Big Data



INFOBRIGHT

Case Study 1: Bango

- Headquartered in the UK, Bango provides technology that powers commerce for businesses targeting the growing market of internet-enabled mobile phone users.
- Bango's products collect payment from mobile users for on-line content and services, and provide accurate analytics back to mobile carriers and content providers about marketing campaigns and user behaviour.
- The world's leading brands, as well as thousands of smaller content providers and developers, use Bango products to more effectively run their mobile businesses.

bango | Mobislim Master Package

Welcome | Payment reports | Analytics reports | App analytics

Get FREE Bango Analytics

- Analytics summary
- Visitors
- Pages
- Traffic sources
- Campaigns & goals
  - Campaigns overview
  - Impressions & clicks
  - Campaign parameters
  - Goal pages
  - Goal comparison
- Payment reports

Export detailed reports

- Traffic
- Analytics links
- Content links

Jan 18 2011 - Jan 18 2011 Time offset: (UTC -8:00)

Export report: XML

Campaign parameters

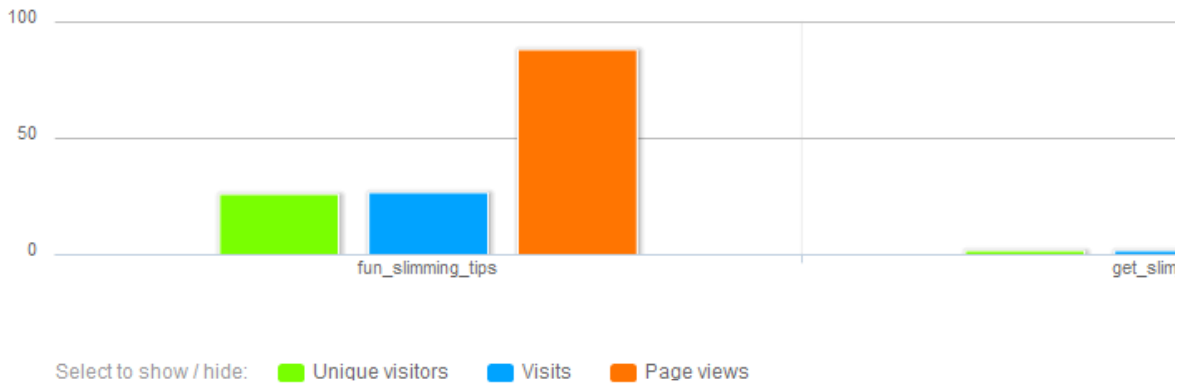
Create a new filte

Totals for date range

Impressions	0
Clicks	0
Unique visitors	28
Page views	102
Visits	29

How we calculate these figures

About this report



NewCampaign | Select parameter...

	NewCampaign	Impressions	Clicks	Unique visitors
1	fun_slimming_tips	0	0	26
2	get_slim_today	0	0	2

Jump to row: Show rows:

Most recent data available for reporting is up to Feb 08 23:59 UTC **49 minutes** ago.

Report cached for **15** minute(s).

Bango Server Time is currently 12

Case Study 2: JDSU

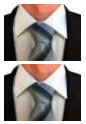
- The JDSU (News - Alert) Session Trace application, and supporting TDR-Store component, ensures quality of service for mobile networks by enabling network operators to quickly identify network issues and pinpoint the source.
- Recognizing the challenge with data growth and the need to contain costs, JDSU set out to deliver a new version of their Session Trace TDR-Store component that would meet the following goals (see next slide):

TDR-Store Requirements

- Support very fast load speeds to keep up with increasing call volume and the need for near real-time data access.
- Reduce the amount of physical disk capacity required by 5x, enabling more data to be stored.
- Significantly reduce overall database licensing costs.
- Eliminate their customers' "DBA tax," which meant that the new solution should require zero maintenance or tuning while enabling flexible analysis.
- Continue to deliver the fast query response needed by NOC personnel when troubleshooting issues and supporting up to 200 simultaneous users.

Case Study 3: DOZ

Senior Management



Sales & Local Coordinators (100 end users)

Analytical Department (6 people)

SQL

Truly ad hoc; often related to immediate requests from senior management; need to be executed in parallel to standard reports

PHP layer built by DOZ

10 concurrent users on average; almost 1TB; a need to reload 200 GB every week

Query results emailed as Excel with embedded SQL

dbam o zdrowie

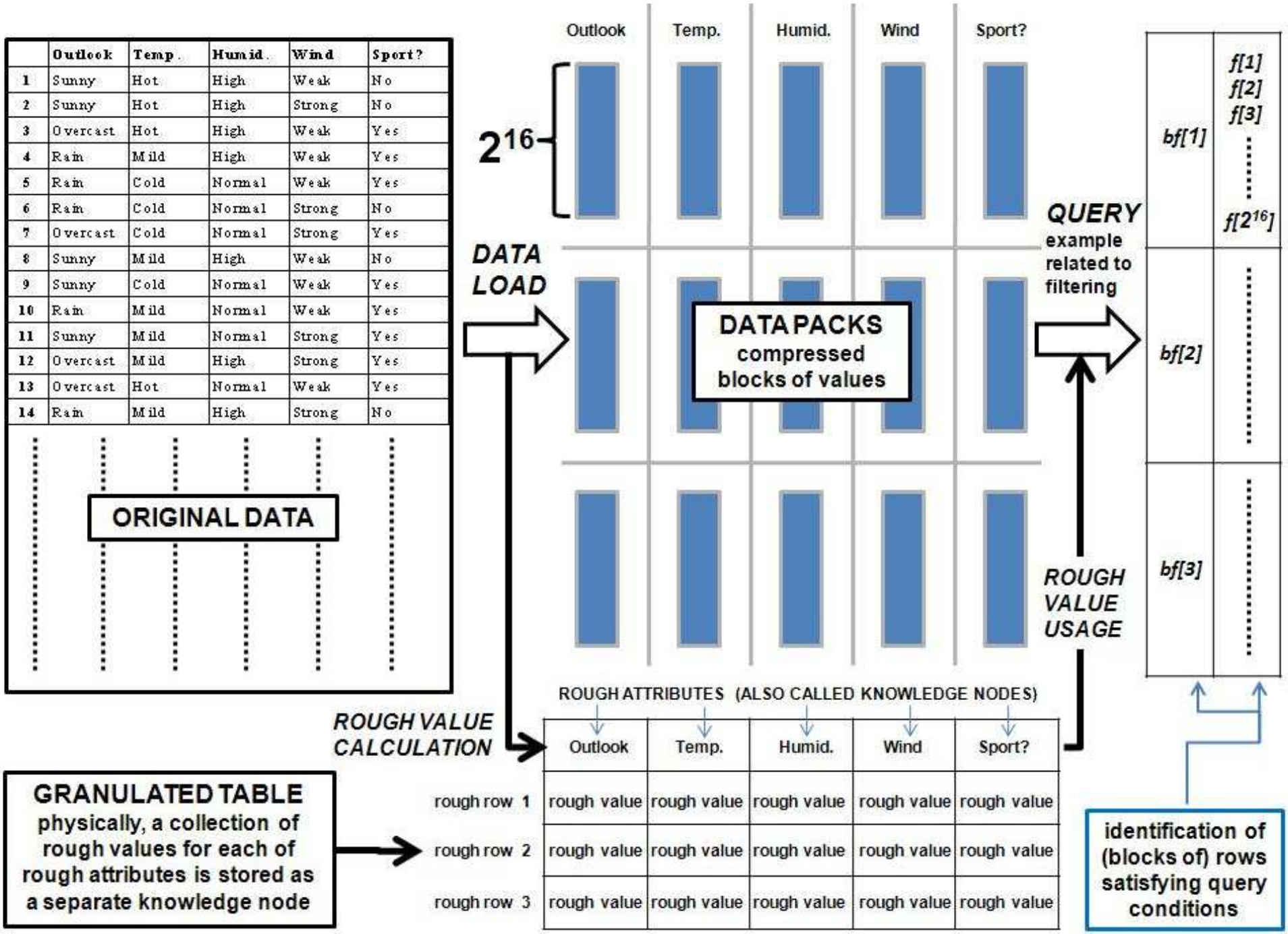
INFOBRIGHT

ICE / IEE: High-Performance RDBMS

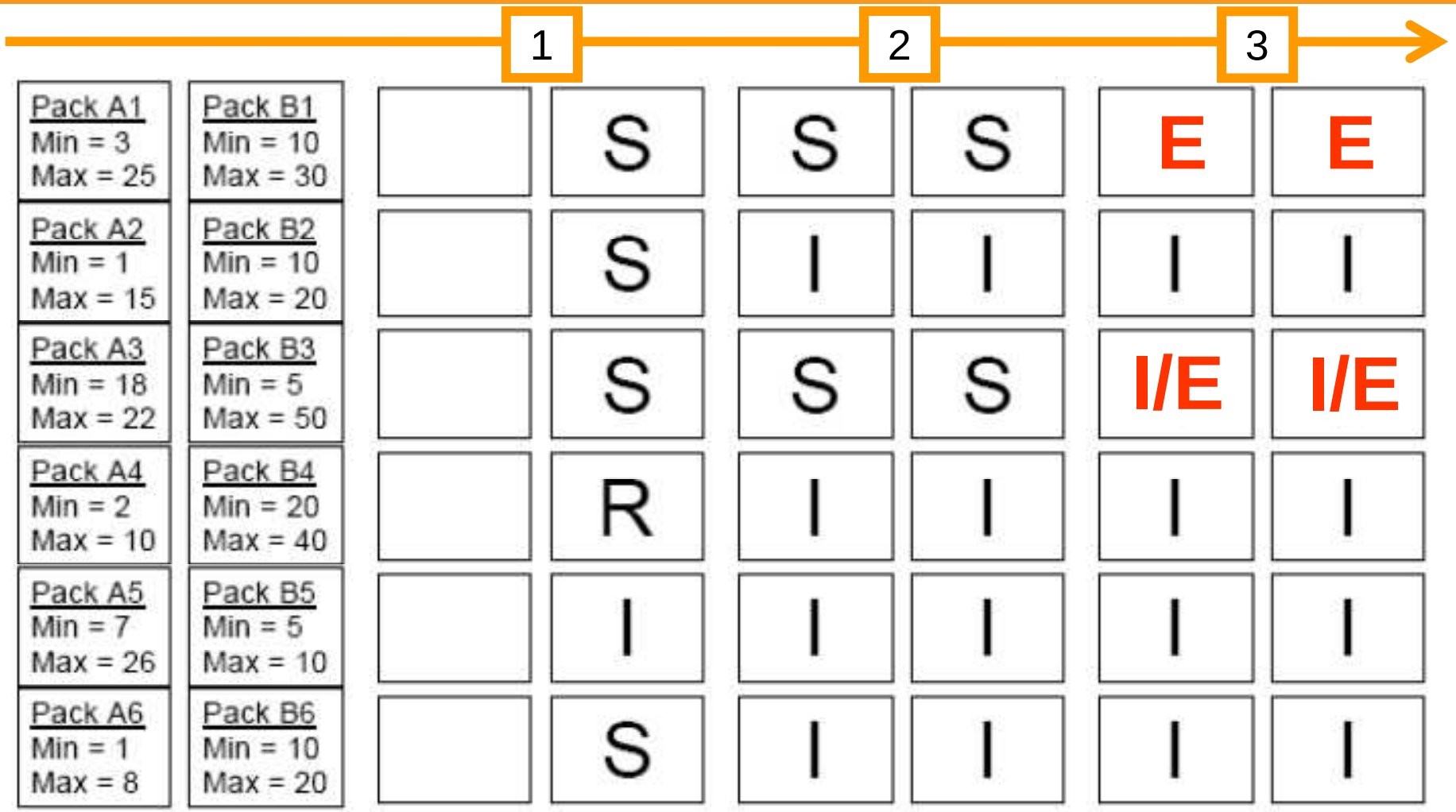
Creates information about the data upon load, automatically	• Stores meta data in the Knowledge Grid (KG) • KG is loaded into memory • Less than 1% of compressed data size
Processing query with metadata = sub-second response time	• Less data that needs to be accessed, faster the response • Eliminate / reduce need to access data when answered by the KG
Reduces IT/DBA Overhead	• No need to partition data, create/maintain indexes, projections or tune for performance • *Ad-hoc* queries are as fast as static queries, so users have total flexibility

More intelligence =
Faster query response and greater rate of compression

INFOBRIGHT


Outlook
Temp.
Humid.
Wind
Sport?
1 Sunny Hot High Weak No
2 Sunny Hot High Strong No
3 Overcast Hot High Weak Yes
4 Rain Mild High Weak Yes
5 Rain Cold Normal Weak Yes
6 Rain Cold Normal Strong No
7 Overcast Cold Normal Strong Yes
8 Sunny Mild High Weak No
9 Sunny Cold Normal Weak Yes
10 Rain Mild Normal Weak Yes
11 Sunny Mild Normal Strong Yes
12 Overcast Mild High Strong Yes
13 Overcast Hot Normal Weak Yes
14 Rain Mild High Strong No
ORIGINAL DATA
2^16
DATA LOAD
DATA PACKS
compressed
blocks of values
QUERY
example
related to
filtering
bf[1]
f[1]
f[2]
f[3]
f[2^16]
bf[2]
bf[3]
ROUGH
VALUE
USAGE
ROUGH ATTRIBUTES (ALSO CALLED KNOWLEDGE NODES)
ROUGH VALUE
CALCULATION
Outlook
Temp.
Humid.
Wind
Sport?
rough row 1
rough row 2
rough row 3
rough value
GRANULATED TABLE
physically, a collection of
rough values for each of
rough attributes is stored as
a separate knowledge node
identification of
(blocks of) rows
satisfying query
conditions

SELECT MAX(A) FROM T WHERE B>15;



I/S/R denotes irrelevant/suspect/relevant; E – exact computation (decompression)

Algorithm 1 TableCheck

Input: *Tables*, *Conditions* $\circledast_T$
Result: *BlockFilters*, *Filters*

```
for all T ∈ Tables do
  for all block ∈ Blocks_T do
    bf_T[block] := 1
    if Rough(block, ⊛_T) = ℑ then
      bf_T[block] := 0
    end if
  end for
end for
RoughProjection(BlockFilters, Conditions)
for all T ∈ Tables do
  changed := FALSE
  for all block ∈ Blocks_T, bf_T[block] = 1 do
    bf_T[block] := 0
    for all row ∈ block do
      f_T[row] := 1
      if Exact(row, ⊛_T) = ℑ then
        f_T[row] := 0
      else
        bf_T[block] := 1
      end if
    end for
    if bf_T[block] = 0 then
      changed := TRUE
    end if
  end for
  if changed then
    RoughProjection(BlockFilters, Conditions)
  end if
end for
```

Algorithm 2 RoughProjection

Input: *BlockFilters*, *Conditions* $\circledast_{T,T'}$
Result: *BlockFilters* further refined

```
done := FALSE
while !done do
  done := TRUE
  for all T, T' ∈ Tables, T' ≠ T do
    for all block ∈ Blocks_T, bf_T[block] = 1 do
      bf_T[block] := 0
      for all block' ∈ Blocks_T', bf_T[block'] = 1 do
        if Rough(block, block', ⊛_T,T') ≠ ℑ then
          bf_T[block] := 1
          break
        end if
      end for
      if bf_T[block] = 0 then
        done := FALSE
      end if
    end for
  end for
end while
```

f[...] and bf[...] represent information about rows and whole blocks of rows that do not need to be processed to resolve a query. In case of blocks it means no need of decompressing corresponding data packs.

Algorithm 3 HashBlockJoin

Input: $bf_T, bf_{T'}, *_{T,T'} \equiv (T.a = T'.a')$
Result: X – pairs of rows satisfying $*_{T,T'}$

```
Initialize(H)
for all block ∈ Blocks_T, bf_T[block] = 1 do
  for all row ∈ block, f_T[row] = 1 do
    if !Full(H) then
      Add((row, a(row)), H)
    else
      for all block' ∈ Blocks_T', bf_T'[block'] = 1 do
        if Rough(H, block', *_T,T') ≠ ℑ then
          for all row' ∈ block', f_T'[row'] = 1 do
            Add({(row, row') : (row, a'(row')) ∈ H}, X)
          end for
        end if
      end for
      Empty(H)
    end if
  end for
end for
```

SELECT ... FROM T INNER JOIN T'
ON T.a = T'.a' WHERE ...;

SELECT B, A* FROM T
GROUP BY B WHERE...;

Algorithm 4 Aggregation

Input: bf_T, aggregates A^*, groupings B
Result: X – the result of aggregation

```
Initialize(H)
done := FALSE
while !done do
  done := TRUE
  for all block ∈ Blocks_T, bf_T[block] = 1 do
    if Full(H) ∧ Rough(block, H, B) = ℑ then
      done := FALSE
    else if Uniform(block, B) then
      if Full(H) ∧ ∀_{h∈H} B(block) ≠ B(h) then
        done := FALSE
      else
        bf_T[block] := 0
        if ∃_{h∈H} B(block) = B(h) then
          for all a* ∈ A* do
            if Rough(block, a*(h)) ≠ ℜ then
              Refresh(block, a*(h))
            end if
          end for
        else
          Add((B(block), A*(block)), H)
        end if
      end if
    else
      for all row ∈ block, f_T[row] = 1 do
        if Full(H) ∧ ∀_{h∈H} B(row) ≠ B(h) then
          done := FALSE
        else
          f_T[row] := 0
```

AND SO ON...

Structured Columns

IP
255.255.255.0
255.0.0.0
10.20.30.15

URI
http://ismis2011.ii.pw.edu.pl
http://www.google.pl/&q=rough+sets&aq=f
ftp://www.host.com/source.zip

postal code
02-078
02-097
00-950

text
Ontology driven concept approximation.
Struktura i funkcja nauk medycznych.
Wisdom technology: A rough-granular perspective.

phone number
+48 225544503
506 999 999

e-mail
dominik.slezak@infobright.com
centralreservation@syrena.com.pl
konferencje@ibib.waw.pl

GPS coord.
52º13'56"N
21º00'30"E

weblog
...

cookie
...

ID
...

INFOBRIGHT

Introducing ICE / IEE 4.x

Built-in intelligence for machine-generated data: Find 'Needle in the Haystack' faster

"DomainExpert"

Intelligence about machine-generated data drives faster performance

- Enhanced Knowledge Grid with domain intelligence automatically optimizes database for specific use cases
- Users can directly add domain expertise to drive faster performance

Near-real time, ad-hoc analysis of Big Data

- Linear scalability of data load for very high performance
- Hadoop connectivity: Use the right tool for the job
- Data mining "drill down" at RAM speed

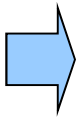
INFOBRIGHT

DomainExpert: Simplified Interface

- Pattern specification enables faster query performance
  - Patterns defined and stored
  - Complex fields decomposed into more homogeneous parts
  - Database uses this information when processing query
- Pre-defined data types for machine-generated data
  - URL
  - E-Mail addresses
  - IP Addresses
- Users can also easily add their own data patterns
  - Identify strings, numerics, or constants
  - Financial Trading example – ticker feed
    “AAPL–350,354,347,349” encoded “%s-%d,%d,%d,%d”

DomainExpert: Simple Example

VARCHAR(100)
„1.2.3.4”
„255.2.2.2”
„255.0.0.0”
„hello world”
„2.2.2.2”
„2..1.0”



meta	%d1	%d2	%d3	%d4	outliers
1	1	2	3	4	„hello world”
1	255	2	2	2	„2..1.0”
1	255	0	0	0	
0	2	2	2	2	
1					
0					

And... How about Approximate SQL?

- In such areas as, e.g., Business Intelligence and Web Analytics, there is an ongoing debate whether the answers to SQL statements have to be always exact.
- The same question occurs in the case of SQL-based machine learning algorithms, which are often based on heuristics, randomness and inexactness anyway.
- Motivation for SQL approximations is related also to such aspects as complexity of queries and data sources, dynamically changing data with a limited access, and huge data sets with a need to monitor convergence of query execution in time, regardless of whether the final answers are to be exact or not.

Rough Query: Speed Up Data Mining

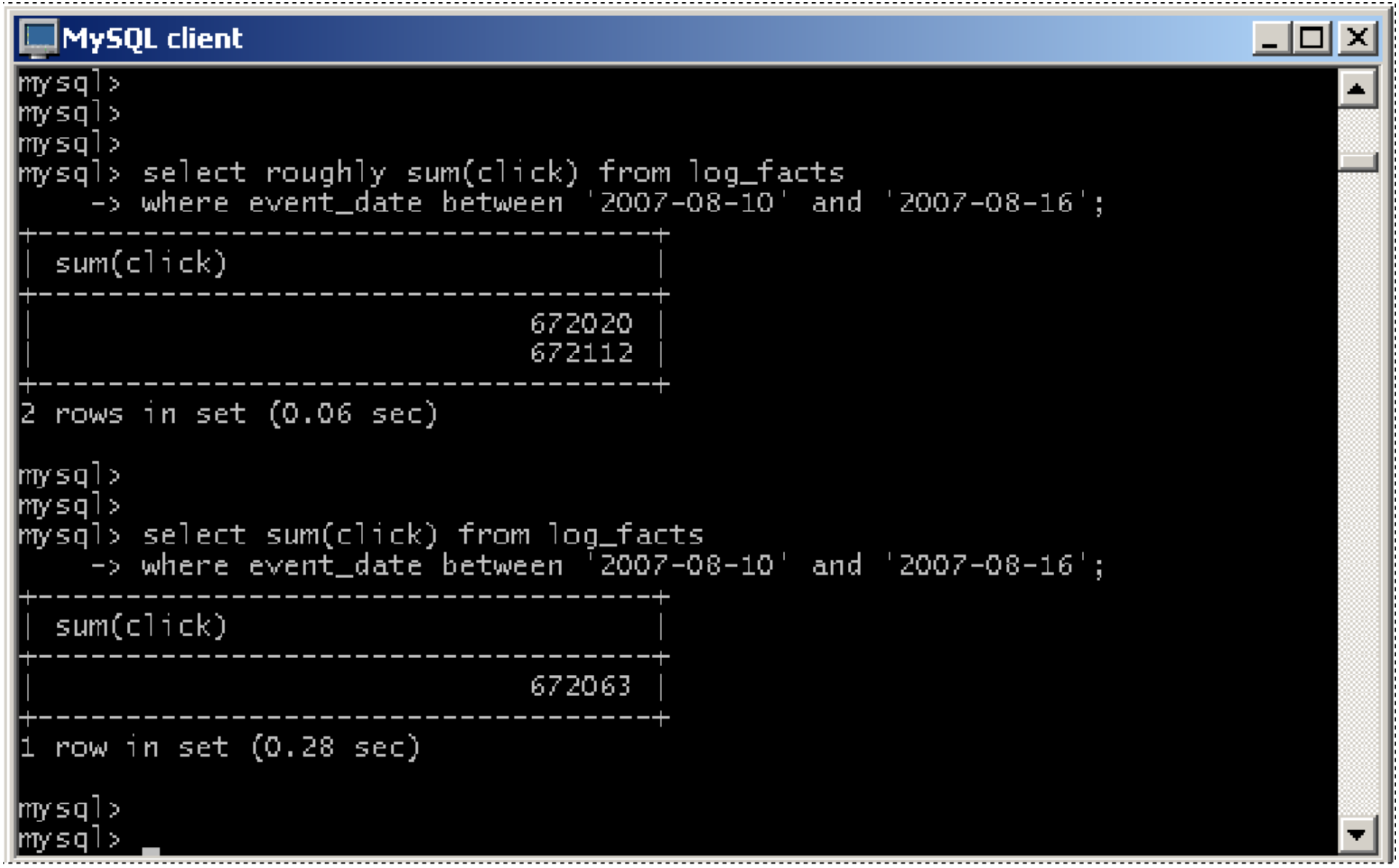
Near-real time ad-hoc analysis

Rough Query: Data mining "drill down" at RAM speed

- Enables very fast iterative queries to quickly drill down into large volumes of data
- "Select roughly" to instantaneously see interval range for relevant data,
  - uses only the in-memory Knowledge Grid information
- Filtering can narrow results
- Need more detail? Drill down further with rough query or query for exact answer

INFOBRIGHT

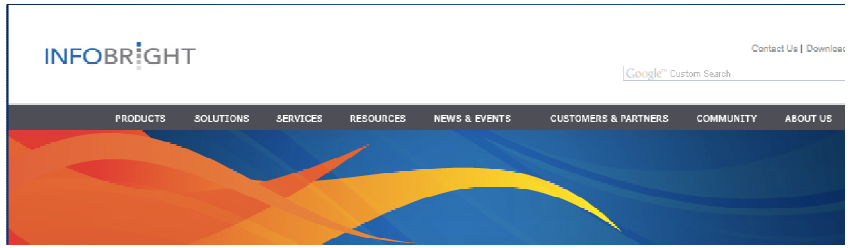
Rough Query: Simple Example



Rough Query: Semantics of Results

- The outcome of an arbitrary SQL statement can be interpreted as an information system with some attributes and objects.
- The outcome of Rough Query corresponds to the collection of statistical snapshots of the values of each of outcoming attributes.
- You can think about it as producing the outcoming rough values from the original rough values stored in our Knowledge Grid.

Thank you!..... slezak@infobright.com



- infobright.org
  - Download ICE (Infobright Community Edition)
  - Download an integrated virtual machine:
    - ICE-Jaspersoft or ICE-Jaspersoft-Talend
  - Join the forums and learn from the experts!
- infobright.com
  - Download a white paper from the Resource library
  - Watch a product video
  - Download a *free* trial of IEE (Infobright Enterprise Edition)

INFOBRIGHT