Is (Your) Database Research Having Impact?

Guy M. Lohman
IBM Almaden Research Center

Outline for a Keynote Speech

- Congratulations on Past Successes!
- Storm on the Horizon!
- Roadmap for Challenges Before Us

More specifically to this talk …

- How does one define “IMPACT”? 
- Have we as a research community had impact? 
- How do we increase our impact? 
- What research had impact…or not … and why? 
- The common factors in successful impact 
- Trends that will affect our future research

If You’re About to Walk Out on This Presentation…

- If you (like me!)
  - Are jet-lagged
  - Have urgent e-mail
  - Would rather be drinking coffee
  - Don’t care for keynotes
  - Or just don’t like the presenter…

- Please remember just one word:

  Con sum a ble (adj.) – able or meant to be consumed, as by eating, drinking, or using

- If we want our research to have impact, it must be consumable:
  - Works “out of the box”
  - Does what I would expect
  - Simple, simple, simple (from the user’s point of view!)
How do you define IMPACT?

Papers are great, …

but are they using your stuff?

Hey, Guy, what about exploratory science and theory? You never know when something will pay off!

Answer #1: I’m primarily addressing systems work.
Answer #2: Theory has a longer time horizon …
   …but I’d prefer impact in my lifetime! ;-)
Answer #3: Newton invented calculus because he needed it, not for its inherent elegance.

Ron Fagin

Are they using our stuff? Yes!

- Relational Databases are a WHOPPING SUCCESS!
  - $14B business in 2004 (double that of 1994!)
  - Run portions of virtually every Fortune 1000 company
  - Ubiquitous & used in critical applications in major enterprises
    - Transactional systems
    - Data warehouses

- Many of today’s products derive from research prototypes
  - Original relational prototypes (System R, Ingres, …)
  - Extensible, Object-Relational prototypes (Postgres, Starburst, Volcano…)
  - Parallel prototypes (Gamma, SDD-1)

So give yourselves a big pat on the back!

BUT…

- How many of you have a digital camera?
- Do you use a (SQL) database to store your photos?
- Why not?
- You are a database expert, aren’t you?
- Answer: It’s too much hassle to set up!
Some more embarrassing questions:

- Why is most (>80%) of the world’s data still not stored in databases?
- Didn’t we “solve” this problem in the 1980s with object-relational DBMSs?
- Have you ever tried to query (with SQL) an existing database that:
  - you didn’t create, and…
  - had more than 500 tables?
  - How long before you got your first SQL query to produce meaningful results?
- Just how easy is it to incrementally add DB capacity beyond
  - 1 machine?
  - 100 machines?
- Do query optimizers really make life easier for DBAs?
- Have “self-managing” databases significantly simplified administration?

How do we increase our impact?

1. Candidly assess
   a. Impact of past research

What past research had impact... or not... and WHY?

- Historically, there have been a few “hot topics” in database research at any one time, e.g.
  - Distributed DBMSs
  - Object-Oriented DBMSs
  - Extensible DBMSs
  - Deductive DBMSs
  - Data Mining
  - XML DBMSs
  - Stream DBMSs
- Which became successfully used, or not, and why?

My Hypothesis:

- Successful Technologies
  1. Were easier to use than previous technologies
  2. Vital to user need (functional, performance)
  3. Were standardized early
- Unsuccessful Technologies
  1. Were more complicated or hard to grasp
  2. Filled a narrow niche at best
  3. Weren’t standardized before products emerged
Case Study 1: Relational DBMSs

- Hot topic in: 1970s
- Customer pain points:
  - Development cost of applications (navigating complex structures)
  - Relationships limited to those specifically stored in the data
- Easier to use than predecessors (e.g., IMS & DL/1)
  - Simple data structure (table)
  - Declarative language (SQL)
- Standardization
  - First official SQL standard in 1986 (query and update, only!)
  - *But de facto* standard of SQL established by earliest products:
    - SQL/DS (1982)
    - Oracle (1983)
    - DB2 (1984)
- Market size: $15.5B in 2006, not including many tools
- My impact assessment thus far: A+

Case Study 2: Distributed DBMSs

- Hot topic in: early to mid-1980s
- Two flavors:
  a. **Parallel DBMSs** – distributed because you *want* to
     - For performance or capacity
  b. **Federated DBMSs** – distributed because you *have* to
     - The data is already distributed into distinct systems

Case Study 2a: Parallel DBMSs

- Forked early on into
  - Special-purpose Hardware: **Database Machines** (e.g., Britton-Lee)
  - General-purpose Hardware: **Parallel DBMSs**
    - e.g., Tandem’s Non-Stop SQL, Teradata, DB2, Oracle
- What Went Right?
  - Addressed critical customer need: **Scalable Performance**
  - Parallelization was hidden from developer (SQL DML largely unchanged)
  - So standardization not critical
- What Went Wrong?
  - Administration of multiple nodes not transparent (complicated)
  - Special-purpose Database Machines couldn’t ride technology wave
- Market Size: $4-6B
- My impact assessment thus far:
  - Parallel DBMS: B+ (most scalable use of data parallelism in the industry!)
  - Database Machines: F

Case Study 2b: Federated DBMSs

- What Went Right?
  - Integrating distinct database systems *ex post facto*
  - Move the data only in response to queries
- What Went Wrong?
  - DBA had to understand and map between diverse schemas
  - Standardization never done
- Market Size: $0.5B
- My impact assessment thus far: C
  - Still hard to set up nontrivial schema mappings
Case Study 3: “Extensible” DBMSs

- Hot topic in:
  - Late 1980s – Early 1990s
  - Sparked by Object-Oriented DBMSs
- What Went Right?
  - Enabled managing the “other” data types
  - Besides short, uniform records of INT and CHAR of traditional relational
  - Included user-defined actions (methods)
- What Went Wrong?
  - More complicated than file system (and slower, too!)
  - Object-relational extensions to SQL not standardized until 1999
  - By then, products had gone their separate ways
  - Independent Software Vendors (ISVs) preferred “least common denominator” (basic) SQL
  - Still had the “impedance mismatch” between programming languages and SQL
- Market Size: $0B
- My impact assessment thus far: D-
  - Incorporated into major products (e.g., Illustra, DB2 LUW), BUT
  - Minimal penetration into applications

Case Study 4: Deductive DBMSs & Datalog

- Hot topic in: Mid-1980s
  - Sparked by interest in Prolog, due to its use by Japan’s 5th-Generation Computer Systems Project (1982)
- What Went Right?
  - Move parts explosions, ancestor/descendant relationships into queries
  - SQL 99 includes recursive queries
- What Went Wrong?
  - Niche market!
  - Datalog not any more natural than SQL to users
- Market size: $0B (Wikipedia: some open source implementations)
- My impact assessment thus far: F

Case Study 5: Data Mining

- Hot topic in: 1990s
- What Went Right?
  - Automatically finds trends & exceptions, without having an hypothesis
- What Went Wrong?
  - Grab-bag of tools – user had to decide
    - Which tool to use?
    - How to use it and interpret results
  - Users trained in statistics, not data mining
  - Never standardized
- Market size: $0B
- My impact assessment thus far: A
  - Academic acceptance:
  - Incorporation into products: B-
  - Use by applications: D-

How do we increase our impact?

1. Candidly assess
   a. Impact of past research
   b. Our strengths and weaknesses
Our Greatest Strengths

- Organizing information, when semantics known
  - Permit reasoning about data aggregates, e.g. OLAP
  - We can help improve precision in Information Retrieval, web search
- Optimizing declarative languages (SQL)
- Exploiting data parallelism
- Aggregation
- Scalable search (indexing)

Our Greatest Weaknesses:

- Simplicity / Ease of use!
  - Administration
  - Keyword search vs. SQL or XQuery
  - Time to Value
    - ER diagrams → Logical schema → Physical schema
- Unstructured, fuzzy information (when semantics not known, e.g. text)

How do we increase our impact?

1. Candidly assess
   a. Impact of past research
   b. Our strengths and weaknesses

2. Partner with other areas who complement us, e.g.,
   - Information Retrieval – discovery of semantics from text
   - Web search – more natural query interfaces
   - Control Theory / Machine Learning – automate administration
   - Hardware guys – ways to make systems
     - Robust
     - Plug-repairable
     - Plug-upgradeable
   - Scientific DBMSs – meta-data
   - Many more…

Example: Annotating Text (IR techniques)

Extract: phone numbers, people’s names, dates, relationships, etc.

**From:** Tom Shelton <tom.shelton@enron.com>

Attached is an update on the referenced matter. Please call me if you have any questions, comments, or suggestions. I can be reached at 713-223-3426. Steve Kean, you may want to mention this to Ken Lay since he was involved in this very early on (a meeting with Mayor Lanier probably in 1996 or so), and may be involved again. Feel free to forward this memo to others who may be interested.

e.g., Tom Shelton’s phone number is 713-223-3426

Moving more towards the model of scientists:

- **Relational Data Bases:**
  - Fixed schema
  - Schema factored out of data
  - Data changes
  - Update in place

- **Scientific Data Bases:**
  - Schema often bundled with data
  - Data fixed
  - NEVER throw out data! → Versions
  - Different ways of looking at that data → schema changes!
  - Heterogeneity of formats
How do we increase our impact?

1. Candidly assess
   a. Impact of past research
   b. Our strengths and weaknesses

2. Partner with other areas who complement us

3. Start with the consumer’s requirements, not yours
   - Get requirements “outside in”
   - Engage with real consumers (including ourselves)
   - LISTEN for what the real problem is
     - Probably goes beyond the “problème du jour”
     - Be careful not to assume away realistic constraints
     - Walk a mile in the consumer’s shoes (use what you build)

Trends that will impact our research

- Increasingly cheap commodity hardware
  - Multi-core chips promise massive CPU parallelism (outdated acronym!)
  - Disk densities improving, but not seek times!
    - 1 TB disk for $399 (Hitachi)
    - Is “disk the new tape”, as Stonebraker observes?
    - Then
      - Memory is the new “disk”
      - Cache is the new “memory”!
  - Why waste a year to get 5-10% improvement in performance?
    - In less time, the hardware improvements will do it for you!

- Consequences:
  1. Massively parallel scale-out
  2. Analyzing Content
  3. Deeper Analytics and “Fishing Expeditions”
  4. People Costs Dominate

Massively Parallel Scale-Out

Trends:
- Data volumes growing fast
  - Cheaper disks permit
    - Larger databases
    - Keeping data longer (compliance!)
    - Keeping old versions (update-in-place is an anachronism)
  - Lots of new kinds of data: RFID, email, photos, videos
- Multi-core chips provide cycles to burn

Requirement #1: Simple & massive scale-out
- Exploit parallelism in 1000s of nodes
- Easily add new capacity
- With low management overhead
- No single point of failure

Analyzing “Content”

Trend: Information converging
- Today – different repositories for different types of data
  - Structured -- traditional Data Base
  - Semi-structured -- traditional Content Management
  - Unstructured (text, multimedia) – file systems
- Each needs a different search interface
  - SQL
  - JSR-170
  - Keyword search / Information Retrieval
- Cheap CPU cycles permit more discovery of text semantics

Requirement #2: Store / Search / Analyze all data
- Need to rapidly relate information of different types
- With one unified interface!
Use Case 1: Customer Sales Opportunities

- What? Integrating...
  - Call center transcripts (text) of what products customer says he has
    - Deployed
    - Problems with
    - Knowledge of
    - Interest in buying
  - Tabular data on:
    - Customer’s history of
      - Purchases
      - Previous calls and problems
    - Product characteristics

- Value:
  - Trends in customer satisfaction with products (product “buzz”)
  - Cheaper & more detailed feedback than product surveys
  - Sales opportunities for cross-sell / up-sell

Use Case 2: Forms Processing

- Examples: Insurance companies (e.g., USAA, Aetna)
- What? Integrating...
  - Forms containing text (may be tagged, or fax images!)
    - Customer forms
    - Claims adjustor (e.g., description of accident)
    - Medical forms (e.g., description of injury)
  - Tabular data on
    - Customers
    - Coverages
    - Incidents

- Value:
  - Reduced time & cost to process
  - More complete information
  - Reduced risk

Use Case 3: Risk Assessment

- Example: Bank (e.g., Citibank)
- What is my company’s exposure to Enron?
  - Tabular data on:
    - Accounts
    - Loans
  - Unstructured text on:
    - Contracts
    - Litigation
    - Correspondence
    - SEC filings
    - Partners who have relationships (transitive)

- Value:
  - More rapid, thorough, & accurate risk assessment
  - Much less labor to assemble

Deeper Analysis & “Fishing Expeditions”

Trend: Awash in data, but not information
- Typical complaint: “I can’t find what I’m looking for!”
- But just finding data isn’t enough!
- Today’s Business Intelligence is too human-intensive
- Cheap cycles permit more compute-intensive analytics

Requirement #3: Pro-actively derive useful information
- Need to automatically
  - Glean more business value from enterprise data
  - Discover the semantics of text
- Need new analytics to exploit unstructured data
- A rebirth of data mining, embedded in DB?
People Costs Dominate TCO

**Trend: Hardware often less than 50% of TCO**
- People costs dominate Total Cost of Ownership (TCO)
- Minimize Time To Value
  - Databases take too long to set up!
- Wizards & Advisors simply mask complexity, add brittleness
- Need to eliminate entire categories of administrative tasks

**Reqmt. #4: System must be simple, robust, & secure**
- Sacrifice resource utilization to **radically** simplify:
  - Setup, Configuration, & Deployment (e.g., **Self-Organizing**)
  - Operation (e.g., problem determination)
- **KISS** – Keep It Simple, Stupid!
- **KIWI** – Kill It With Iron [Weikum]!
- Example: “Good enough” plans exploiting massive parallelism

Research Topics that WILL Have Impact

- **Tools that simplify**
  - Program Development (esp. the “Impedance Mismatch” with DBMS)
  - Database Administration
- **Anything that automatically discovers, in unstructured data,**
  - Semantics
  - Structure
  - Metadata
- **Anything that manages and/or exploits**
  - Metadata
  - esp. Versions
  - Massive Parallelism

Conclusions

- Relational databases changed the world
- Researchers had a major role
- **But …**
  - Some of our research has had little impact (thus far)
  - We can do much better!

- **The key to impact is**
  - Engage with customers, and **LISTEN!**
  - Use it yourself -- Go home and build an application!
  - **KISS & KIWI**
- It’s the PEOPLE, stupid!
- Many interesting problems arise from real-world problems!