Administrivia

- **Final Exam**
  - Saturday, Dec. 15, 8-11A (sorry!)
  - Location: 4 LECONTE
  - Cumulative, stress end of semester
  - 2 cribsheets
- **Final Review Session**
  - Thursday 12/12?
  - Exact time TBA.
    - will blog the time and place

Office Hours

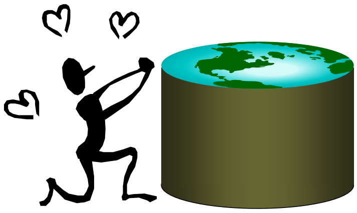
- **This week**
  - Eirinaios OH moving to Thurs @ 5PM, 711 Soda
- **Next week**
  - Prof H office hours as usual Tues/Thurs
  - Additional Prof H office hours Fri 10-2
  - TA office hours TBA

As you study...

- "Reading maketh a full man; conference a ready man; and writing an exact man."
  -Francis Bacon
- "If you want truly to understand something, try to change it."
  -Kurt Lewin
- "I hear and I forget. I see and I remember. I do and I understand."
  -Chinese Proverb.
- "Knowledge is a process of piling up facts; wisdom lies in their simplification."
  -Martin H. Fischer

Database Lessons to Live By

"If we do well here, we
shall do well there:
I can tell you no more
if I preach a whole
year"
-- John Edwin
(1749-1790)



Recall Lecture 1!!

- **Lessons of Data Independence**
  - High-level, declarative programming
  - Maintenance in the face of change
    - Automatic re-optimization
- **Data integrity**
  - Declarative consistency (constraints, FDs)
  - Concurrent access, recovery from crashes.

Simplicity is Beautiful

- **The relational model is simple**
  - simple query language means simple implementation model
    - basically just indexes, join algorithms, sorting, grouping!
  - simple data model means easy schema evolution
  - simple data model provides clean analysis of schemas (FD's & NF's are essentially automatic)
  - Every other structured data model has proved to be a wash
    - XML has found a niche, but not as a database
    - There's a reason that the backend of web search looks so much like a relational database.

Bulk Processing & I/O Go Together

- **Disks provide data a page at a time**
- **Databases deal with data a set at a time**
  - sets usually bigger than a page
  - means I/O costs are usually justified.
  - much better than other techniques, which are "object-at-a-time"
- **Set-at-a-time allows for optimization**
  - can do bulk operations (e.g. sort or hash)
  - or can do things tuple-at-a-time (e.g. nested loops)

Optimize the Memory Hierarchy

- **DBMS worries about Disk vs. RAM**
  - spend lotsa CPU cycles planning disk access
  - I/O cost "hides" the think time
- **Similar hierarchies exist in other parts of a computer**
  - various caches on and off CPU chips
  - less time to spare optimizing here
- **Change is happening here!**
  - Disk is the new tape
  - Flash is the new disk
  - RAM is the really big

Query Processing is Predictable

- **Big queries take many predictable steps**
  - unlike typical OS workloads, which depend on what small task users decide to do next
- **DBMSs can use this knowledge to optimize**
  - For caching, prefetching, admission control, memory allocation, etc.
- **These lessons should be applied whenever you know your access patterns**
  - again, especially for bulk operations!

- **disclaimer**
  - I have done a lot of research based on the contrary
  - but remember: religion first

Applied Algorithm Analysis

- **Know the practical costs of your algorithms**
  - The optimizer needs to know anyway
  - E.g. sorting is not O(*n* log*n*), it's linear
    - 2 passes? 3? 4?!
- **In many applications, the bottlenecks determine the cost model**
  - e.g. I/O is traditional DB bottleneck
  - in another setting it might be network, or processor cache locality
  - this affects the practical analysis of the algorithm

Indexing Is Simple, Powerful

- **Hash indexes easy and quick for equality**
  - worth reading about linear hashing in the text
- **Trees can be used for just about anything else!**
  - each tree level partitions the dataset
  - labels in the tree "direct query traffic" to the right data
  - "all" you need to think about in designing a tree is how to partition, and how to label!

Not enough memory? Partition!

- **Traditional main-memory algorithms can be extended to disk-based algorithms**
  - partition input (runs for sorting, partitions for hash-table)
  - process partitions (sort runs, hash partitions)
  - merge partitions (merge runs, concatenate partitions)
- **Sorting & hashing very similar!**
  - their I/O patterns are "dual"

Declarative languages are great!

- **Simple: say what you want, not how to get it!**
- **Should correctly convert to an imperative language**
  - Codd's Theorem says rel. calc. = rel. alg.
  - no such theorem for text ranking :-(
- **If you can convert in different ways, you get to optimize!**
  - hides complexity from user
  - accomodates changes in database without requiring applications to be recompiled.
- **Especially important when**
  - App Rate of Change << Physical Rate of Change
- **A reborn trend in computing**
  - Declarative networking, security, robotics, natural language processing, distributed systems, …

SQL: The good, the bad, the ugly

- **SQL is very simple**
  - SELECT..FROM..WHERE
- **Well…SQL is kind of tricky**
  - aggregation, GROUP BY, HAVING
- **OK, OK. SQL is complicated!**
  - duplicates & NULLs
  - Subqueries
  - dups/NULLs/subqueries/aggregation together!
- **Remember: SQL is not entirely declarative!!!**
- **But, it beats the heck out of writing (and maintaining!) C++ or Java programs for every query**

Query Operators & Optimization

- **Query operators are actually all similar:**
  - Sorting, Hashing, Iteration
- **Query Optimization: 3-part harmony**
  - define a plan space
  - estimate costs for plans
  - algorithm to search in the plan space for cheapest
- **Research on each of the 3 pieces goes on independently! (Usually…)**
- **Nice clean model for attacking a hard problem**

Database Design

- **(And you thought SQL was confusing!)**
- **This is not simple stuff!!**
  - requires a lot of thought, a lot of tools
  - there's no cookbook to follow
  - decisions can make a *huge* difference down the road!
- **The basic steps we studied (conceptual design, schema refinement, physical design) break up the problem somewhat, but also interact with each other**
- **Complexity in DB design pays off at query time, and in consistency**
  - vs. files

CC & Recovery: House Specialties

- **RDBMSs nailed concurrency and reliability**
  - transactions & 2-phase locking
  - write-ahead-logging
  - details are tricky, worked out over 20 years!
- **Also models for relaxing transactions**
  - Lower degrees of consistency
- **Other systems are now taking pieces**
  - Journaling file systems
  - Transactional memories
  - Web infrastructure locking services (Chubby)

The Rebirth of Information Retrieval

- **A lonely backwater in the 70's, 80's, early 90's**
- **Now a driver of research and industry**
- **We saw that it's easy to get working**
  - But there's tons more!
  - Watering hole for ideas from databases, AI, approximation algorithms, distributed systems, power-efficient processors, HCI, …
  - Kicking off the new generation of parallel dataflow
- **Pushing to yet another level of scalability**
  - Always a game-changer

Databases: The natural way to leverage parallelism & distribution

- **The promise of CS research for the last 15 yrs:**
  - There are millions of computers
  - They are spread all over the world
  - Harness them all: world's best supercomputer!
- **This was routinely disappointing**
  - except for data-intensive applications (DBs, Web)
- **2 reasons for success**
  - data-intensive apps easy to parallelize & distribute
  - lots of people want to share data
  - fewer people want to share computation!
- **The parallelism craze is BACK**
  - Intel, AMD, etc *need* us to take advantage of parallelism
    - They have nothing else to do with all those transistors!
  - Google convinced people that bulk data analysis is cool
    - Map/Reduce
    - Incoming freshman will get this in 61A and through the curriculum

"More, more, I'm still not satisfied"

-- Tom Lehrer

- **Grad classes @ Berkeley**
  - CS262A: a grad level intro to DBMS and OS research
    - Next spring, Hellerstein/Brewer
  - CS286: grad DBMS course
    - Next year, Franklin
  - read & discuss lots research papers
    - See evolution of different communities on similar issues
  - undertake a research project -- often big successes!
- **CS298-12 (aka DB Lunch)**
  - Fridays at 12:30
- **Upcoming seminar courses**
  - I will also offer a 294 this spring
  - Alon Halevy from Google will offer something in Fall '08

But wait, there's more!

- **Graduate study in databases**
  - Used to be rare (Berkeley + Wisconsin)
  - You are living in the golden age:
    - Berkeley, Wisconsin, Stanford, MIT, Brown, Cornell, CMU, Maryland, Penn, Duke, Washington, Michigan, many others...
- **Tons of DB-related companies, lots of hiring**
  - Search companies
  - DB "elephants" : IBM, Oracle, MS
  - Midstage DB startups: ANTs, Greenplum, Netezza
  - Early startups: Truviso, Streambase, Coral8, Vertica, Paraccel …
  - Enterprise app firms: e.g., SAP, Salesforce
  - Every Web 2.0 company!
- **A note: ask for the job you want**
  - E.g. not just engineering -- sales, marketing, R&D, management, etc.

Parting Thoughts

- "Education is the ability to listen to almost anything without losing your temper or your self-confidence."
  -Robert Frost
- "It is a miracle that curiosity survives formal education."
  -Albert Einstein
- "Humility...yet pride and scorn;
  Instinct and study; love and hate;
  Audacity...reverence. These must mate"
  -Herman Melville
- "The only thing one can do with good advice is to pass it on. It is never of any use to oneself."
  -Oscar Wilde