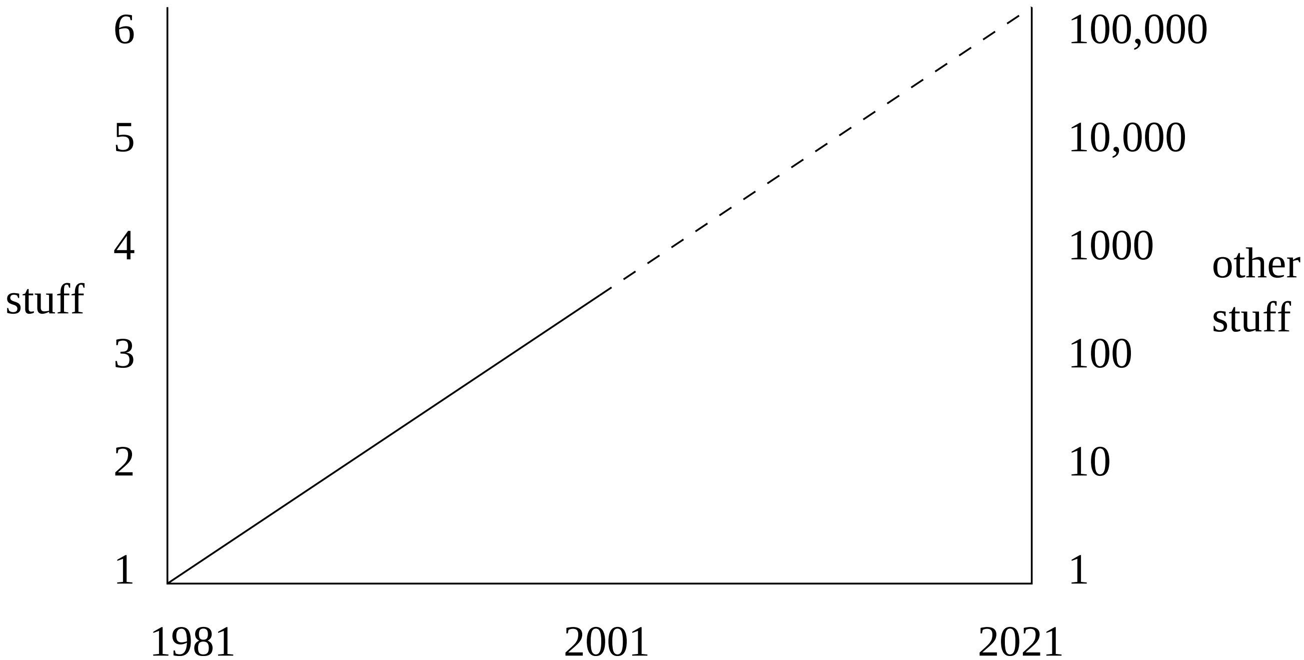
Questions for Today

- What will database system research be like 20 years from now?
  - Rephrase: How do you define “database research” so it makes sense in 2001$\pm$20?
- What will education be like 20 years from now?
  - Rephrase: What can we do to make sure education 20 years from now is not the same as it has been for the past 800 years?

Summary of Talk



- Even this much is hard to get.
  - COSERS (1980) could only get as far as "some day machines will be 10 times as fast as they are now."

Limitations on Predicting the Future

- I believe it is a fundamental fact of living in a technology-driven field that we cannot rely on forecasts of more than a few years.
- Technology changes too fast for it to make sense to devote a lot of intellectual resource to solving a problem that will arise 20 years in the future — it may never appear.
  - Old example: Josephson junctions (a superconducting-based switching technology that was made pointless by IC's).
  - New example: Quantum computers. Theoreticians have developed very detailed computational models for a device that may well prove unbuildable.
- But *basic* research — investivations that are preliminary to a real product — are great as long as there is some reason to believe it will be useful before the next technology shift.

Long Term, What Does a "Database Person" Care About?

- What is the largest amount of data with which we can deal?
  - What can we do with it?
  - How?
- Consequence: the Web is an appropriate domain; so are scientific DB's, image DB's, libraries.
- No apology is needed if we switch from Employees and Departments.

Long-Lived Themes

- High-level query languages.
  - If you are going to deal with large data, there has to be some uniformity, allowing high-level, broadly acting operations.
- Optimization.
  - High-level expression is impossible without good implementation behind the scenes.

What Does Moore's Law Tell Us?

- Lots of things grow by a factor of 10 every 10 years: processor speed, disk capacity, communication capacity, etc.
  - Unless the law fails to hold.
- Other things grow, but more slowly: RAM speed, disk rotation, etc.
- And some things don't grow at all: speed of light, human brainpower, etc.

Interesting Changes if Moore's Law Survives

- In 2021 a high-end computer has a 200GHz processor and a 10Tb disk.
- But in processor cycles, it takes 50 times longer to get anything from disk.
  - Locality, memory hierarchies become even more important, e.g., recent work on B-trees from main-memory to cache.
- Communication rate to the home/office approximates the disk-transfer rate.
  - Will there be a need for local disk at all?

And if Moore's Law Is at an End?

- "They" are now saying 13 years is the maximum Moore's law can survive.
  - Well 13 years ago, it was given 8 years, so who knows?
- On the other hand, there are reasons to believe that the processor-speedup curve is tailing off.
  - Wires are now higher than they are wide (capacitance dominates?).
- Improvements need to come from elsewhere.
  - Parallelism, distribution.
  - Specialized architectures.
  - New algorithms and strategies (e.g., search as a peer-to-peer activity).

Some Nearer-Term Research Targets

Definitely not an exhaustive list.

1. Languages and systems for automating the process of integrating databases (and applications?).
   - Everyone acts as if the problem were solved, but integration is still a bottleneck.
2. Stream-management systems.
   - There are lots of applications where data whizzes by so fast it is almost impossible to process in real time: telecom billing, intrusion detection, Web trends, ...
3. Data mining.
   - Is there anything beyond the a-priori algorithm with general utility?
4. Automation of database design and tuning: no more DBA.
5. Retrieval of new kinds of data, e.g., images.

6. All sorts of optimization questions:
   (a) Object-oriented data.
   (b) Semistructured data/XML.
   (c) Optimization of queries over integrated data: capability-based and cost-based optimization.
   (d) Optimization to take advantage of an extensive memory hierarchy.

Possible CS Research Futures

1. All research on efficient processing of large-scale data is deemed to be a threat to copyright and violates the DMCA. There are arrests at SIGMOD conferences, and DB research is halted.
2. Federal funding of CS research shrinks, and only applications to science are supported. Because scientists see only numerical computation as important, DB research receives essentially no federal funding.
3. Federal funding for academic research is seen to be social and demographic rather than technical. Small amounts of research money are distributed with a salad shooter.

What Will Save the Situation?

- I actually believe that all three of these trends are real, but (1) may get shot down if the Supreme Court displays its usual great wisdom. :-)
- One possible "save": Research doesn't cause wealth; wealth causes research.
  - Companies like Microsoft and Oracle start putting significant money into basic research.

- Another possibility: Venture capital leads the way:
  - The dot-bomb phenomenon should remind everyone involved that *smart technologists* are needed, not MBA's with a "business plan."
  - There is a 20-year pattern of steady growth in venture capital and information-technology startups, upon which a boom-bust cycle has recently been imposed.
  - Look for the long-term trend to resume after the shakeout.
  - It is in the interest of the VC community, more than tradational companies, to support bright, young kids learning and inventing.
  - VC support is merit-based (well, at least along one dimension of "merit"), and therefore optimizes the amount of achievement for a $ invested.

Destroying Education as We Know It

- Education seems to resist the cost improvements that have dominated almost every other industry.
- On-line education is failing — doesn't address the needs of more than a small percentage of learners.
- Education research has done some things to improve *quality*, not *cost*.
  - On-line demos, on-line literature and databases, etc.

Another Approach

- Accept that residential colleges are a good thing.
  - Students learn more out of the classroom than in.
- Eliminate costs by eliminating redundancy, while retaining those aspects of education that need to be one-on-one or one-on-small-number.
- Support instructors, and let them handle more students, rather than replacing instructors.

Database Course As an Example: What Can Be Centralized?

1. Lectures: large choice of topics already prepared.
2. Assignments and grading.
3. Certain kinds of assistance to students: debugging, “can we assume in HW problem 2 that $\cdots$?”
   - 24/7 hotline to reach a distributed pool of TA’s.
4. A professionally managed DBMS.

The New Course Model

1. School assigns instructor to a course, but contracts for services with a service provider.
2. Instructor manages course and provides “valued-added” services, such as small discussion groups, office hours.
3. Instructor also administers exams and assigns grades, based on local policy and the scores reported by the service provider.

Real Summary

- We need to follow the big-data issues, wherever they arise.
- Memory hierarchy issues become more important.
- Optimization in its various guises remains central.
- Database systems is a good poster child for reducing the cost of education.