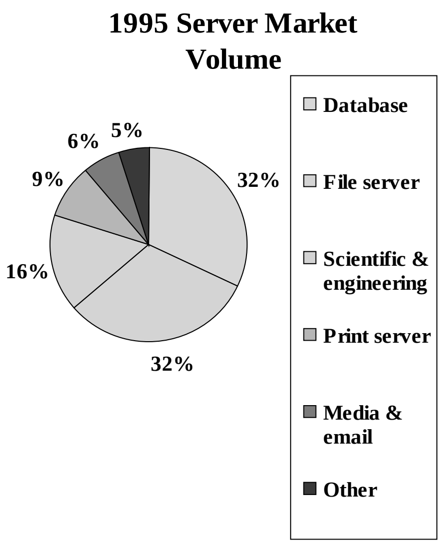
The Intelligent Disk (IDISK): A Revolutionary Approach to Database Computing Infrastructure

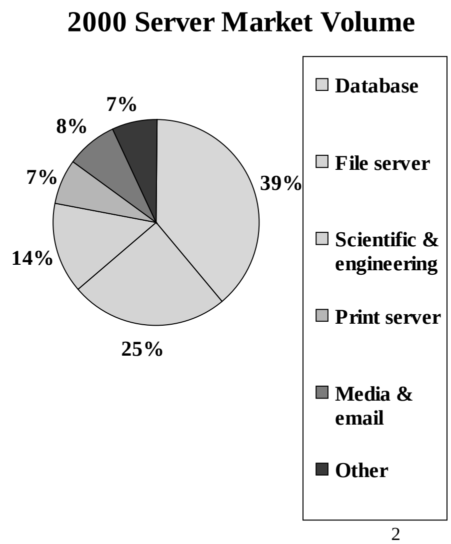
Kimberly Keeton, David A. Patterson, Joseph Hellerstein, John Kubiatowicz, and Katherine Yelick

UC Berkeley Computer Science Division
{kkeeton,patterson,jmh,kubitron,yelick}@cs.berkeley.edu
http://iram.cs.berkeley.edu/

Motivation: Server Market Breakdown

Source: Stenstrom, et al., *IEEE Computer*, December 1997





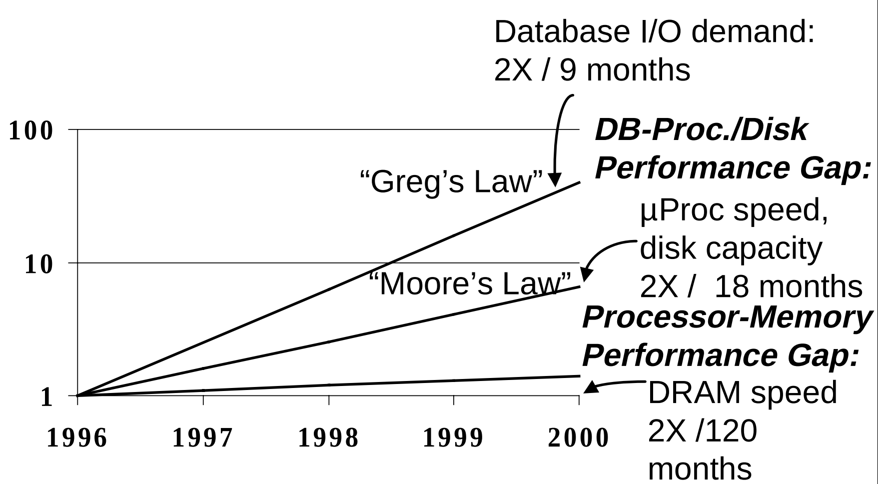
2

Motivation: Decision Support Databases

Characteristic	**DSS**	**OLTP**
Business question	Historical: support for forming business decisions	Operational: day-to-day business transactions
Industry benchmark	TPC-D	TPC-C
Query complexity	Long, very complex queries	Short, moderately complex queries
Portion of DB accessed per query	Large	Small
Type of data access	Read-mostly	Read-write
How are updates propagated?	Periodic batch runs or background “trickle” streams	Through most transaction types

3

Motivation: Database Demand vs. Processor/DRAM speed and Disk Capacity



4

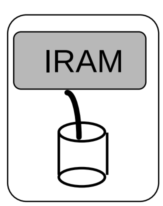
Motivation: Increasing Compute and I/O Needs

- Greg's Law: Greg Papadopoulos, CTO, Sun Microsystems
  - DSS database I/O demand growth: 2X / 9 months
  - I/O capacity and associated processing
- Contributing factors:
  - Collect richer data (i.e., more detailed)
    - "Just-in-time" inventory: connect sales to suppliers
  - Keep longer historical record
  - Growth of digital data
  - Business consolidation
- Winter VLDB Survey (1997):
  - Telecomm., retail & financial DBs ~doubled from 1996 to 1997
  - "Wal-Mart says that a major obstacle to its VLDB plans is that hardware vendors can barely keep up with its growth!"

5

Motivation: "Intelligent Disk" (IDISK)

- IDISK: Processor+memory+fast network per disk
- Push processing to data, rather than data to CPU
- Allows processing of system to scale with increasing storage demand
- Fast network allows direct IDISK-to-IDISK comm.
- Trades expensive central processor MIPS for less expensive disk processor MIPS



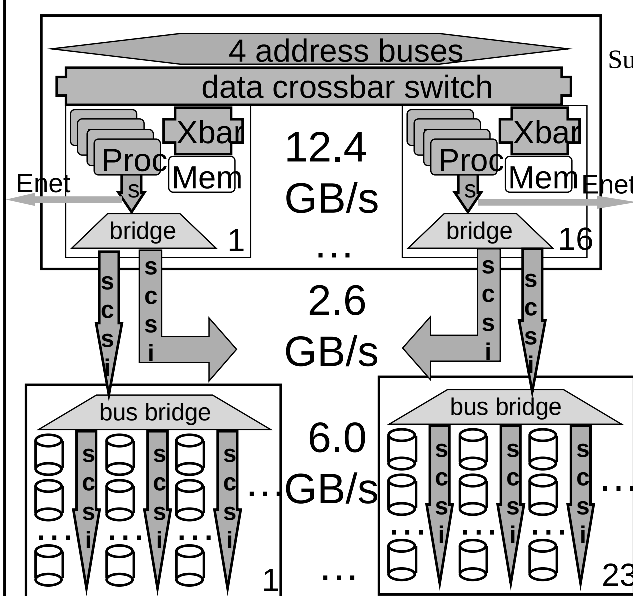
6

Outline

- ✶ Decision support DB servers today
- ✶ Computer architecture trends
- ✶ IDISK proposal for decision support databases
- ✶ Case study: TPC-D Q1
- ✶ Conclusions

7

Current Decision Support Server Architecture



Sun Enterprise 10k (Oracle 8):

- 1 TB TPC-D result
- SMP: 64 CPUs, 64GB DRAM, 603 disks

Disks,encl.	$2,348k
DRAM	$2,328k
Boards,encl.	$983k
CPUs	$912k
Cables,I/O	$139k
Misc.	$65k
HW total	$6,775k

8

Limitations of Current DSS Architecture

- ✶ Experimentally, central CPUs are the bottleneck
  - Processing, storage capacity don't scale easily w/ Greg's Law
- ✶ Desktop processors not tailored to DB applications
  - Somewhat better memory system behavior than TPC-C
    - CPI for TPC-D query 1 ~1.35 (Pentium Pro)
    - CPI for TPC-C ~3.39 (Pentium Pro)
- ✶ Limited I/O bus growth rates
- ✶ Memory system performance: bandwidth and latency
- ✶ Expensive!
  - Central processors and dense memory
  - Cabinets and plumbing handle max configuration

9

Execution Characteristics of TPC-D Workload

- ✶ Experimentally, central CPUs are the bottleneck
  - Somewhat better memory system behavior than TPC-C
    - CPI for TPC-D query 1 ~1.35
    - CPI for TPC-C ~3.39
- ✶ Typical operations
  - Scan, aggregates (e.g., min, max, count) & 1-pass, 2-pass sort/join
  - Avg. 500-2000 instructions/record (DBMS-, query-specific)
  - 200 B/record; <= 6 tables/query
  - Records are read 1-2x, written 0-1x
- ✶ I/O pattern tends to be sequential
  - 8KB - 4MB reads; 8KB - 64KB writes (DBMS-specific)
  - Index scan may be more random (DBMS-specific)
- ✶ Full storage requirements about 2-5x database size

10

Architecture Today: Disk Trends

- Increased disk-resident memory
  - Ex: Seagate Cheetah drive: 1 MB RAM, 4 MB optional
- Increased disk-resident processing
  - ASIC for ECC, SCSI
  - General purpose processor next?
    - (see NSIC/NASD)
  - Intel’s Intelligent I/O ($I_2$0) initiative
- Fast serial lines replacing busses
  - Fibre Channel Arbitrated Loop (FC-AL), Serial Storage Architectures (SSA)
  - Intel’s Gbit/sec serial bus s follow-on to 64b, 66 MHz PCI
- More modularized disk design

11

Architecture Today: Communication Trends

- Serial communication advances
  - Fast (Gbps) serial I/O lines [YangHorowitz96], [DallyPoulton96]
    - State of the art: 4 - 5 Gbps
  - Standardized I/O devices
- Switched networks overtake bus-based networks
  - Switched Ethernet, ATM, Myrinet
  - Fast single-chip switches
    - State of the art: 16-way, GHz / link [Seitz98]

12

Architecture Today: Processor Trends

- ✶ Processors designed for desktop
  - Desktop processors have volume
  - Servers also use desktop MPUs
- ✶ Desktop processors targeted towards SPEC, Windows apps
- ✶ Desktop processors less effective for database workloads
- ✶ Rise of the embedded processor
  - Embedded vs. desktop Dhrystone, SPECint95: < 2x differences
    - Ex: R5000 vs. R10K:
      - SPECInt95 ratio: 1.6
      - Dhrystone ratio: 1.0
  - Fraction of cost
  - Order of magnitude lower power
- ✶ Integrated logic and DRAM on same chip
  - Mitsubishi, LSI Logic, NeoMagic
  - Berkeley's IRAM project

13

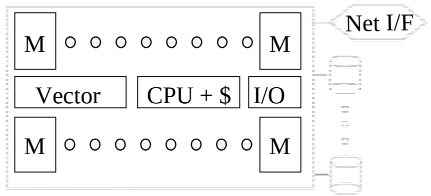
Embedded vs. Desktop Processors

Source: *Microprocessor Report*, Summer 1997

Processor	**Digital SA-110**	**MIPS R5000**	**MIPS R10000**	**Digital 21164**	**Intel Pentium II**
Clock rate	233 MHz	200 MHz	200 MHz	600 MHz	300 MHz
Cache size	16K/16K	32K/32K/ 512K	32K/32K/ 4M	8K/8K/96K/ 2M	16K/16K/ 512K
IC process	0.35 µ 3M	0.35 µ 3M	0.35 µ 4M	0.35 µ 4M	0.28 µ 4M
Die size	50 mm^2	84 mm^2	298 mm^2	209 mm^2	203 mm^2
SPEC95 base (i/f)	n/a	4.7/4.7	10.7/17.4	16.3/19.9	11.6/6.8
Dhrystone	268 MIPS	260 MIPS	203 MIPS	920 MIPS (est.)	n/a
Power	0.36 W	10 W	30 W	25 W	30 W
Est. mfrs. cost	$18	$25	$160	$125	$90

14

Berkeley IRAM Target Parameters



Characteristic	IRAM-I (1999)	IRAM-II (2002)
DRAM Generation	256 Mbit	1 Gbit
On-Chip Memory (MB)	24	96
On-Chip Memory B/W (GB/s)	50 – 200	50 – 200
I/O B/W via N Serial Lines (GB/s)	0.5 – 2.0	0.5 – 4.0
Individual Serial Line B/W (GB/s)	0.25	0.5
On-Chip Memory Latency (ns)	20 – 30	20 – 30
Processor Speed (MHz)	300 – 500	500 – 1000
Vector Performance (GFLOPS)	4	16

+ High on-chip memory B/W, low on-chip memory latency
- Small on-chip memory capacity

15

Proposal: "Intelligent Disk" (IDISK)

- Processor+DRAM+fast network per disk: IRAM!
- Move function to data instead of data to CPU
  - Traditional relational operators: scan, sort, join, …
  - Newer object-relational operators: image, audio, …
  - Other functionality: reorganize multi-dim. data, DB loading
- Potential benefits for DSS databases:
  - Offloads processing from overutilized CPU to disk processors
  - Reduces data movement through I/O system
  - Allows processing of system to scale with increasing storage demand
- Prediction: IDISK variant can be future commodity disk part/option
  - Need to demonstrate high performance at low cost, power

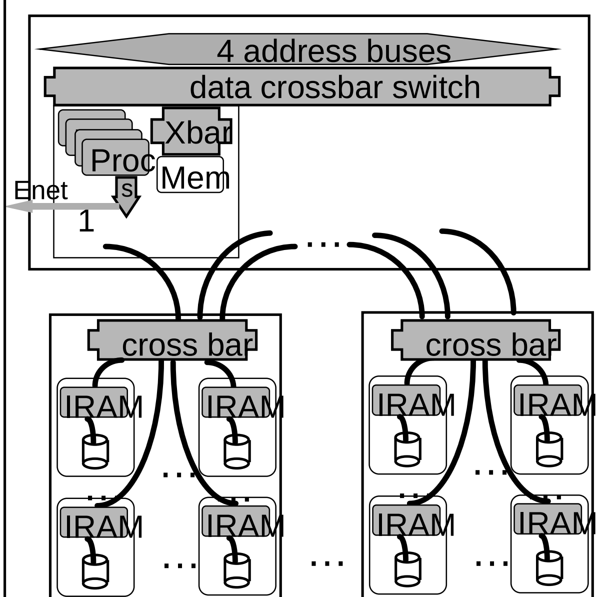
16

Disk Processing for Databases Not a New Idea

- Late 70s, early 80s: database machines
- Several flavors: central host processor +
  - Processor per {head, track, disk}
  - Multiprocessor cache
- After much enthusiasm, failed because:
  - Didn't use commodity hardware
  - Tremendous performance gains for scans didn't justify cost
  - Provided performance improvements only for simple operations (e.g., scans), but not for more complicated operations (e.g., joins)
- Why should IDISK succeed now?
  - Disk with processor+memory can be commodity part
  - Algorithmic advances of last 15 - 20 years
    - Parallel, cluster-based ("shared nothing") join, sort algorithms

17

Evolutionary IDISK Architecture (ISTORE)



- 1 IRAM/disk
- I/O interconnect: crossbar with serial lines
- Trade inexpensive IDISK processing for expensive central MPU processing
- Retain centralized processing
  - Simplify programming model
  - Accept and optimize user queries
  - Assists in executing queries too complex for IDISK alone

18

IDISK Software Architecture

- What is software model for IDISK? Alternatives:
  - Run complete DB server + OS on each disk processor
  - Run all of storage/data manager on each disk node
  - Run small portion of storage manager on each disk node (*)
    - Each disk contains library of kernel operations (scan, join, etc.)
    - Download "arbitrary" user code
    - Use secure programming environment (e.g., Java)?
- Leverage algorithms that trade I/O bandwidth for memory capacity
  - How well will these work in very memory-constrained environment?

19

Case Study: Future SMPs

SMP Characteristic	**1 TB System**	**3 TB System**	**10 TB System**
Processors	64 * 1000 MHz	64 * 1000 MHz	64 * 1000 MHz
Memory capacity	64 GB	256 GB	256 GB
SMP Interconnect B/W	30,000 MB/s	30,000 MB/s	30,000 MB/s
Memcpy B/W	15,000 MB/s	15,000 MB/s	15,000 MB/s
Disk capacity	200 * 36 GB	600 * 36 GB	1800 * 36 GB
Disk transfer rate	29 MB/s	29 MB/s	29 MB/s
I/O interconnect	16*2*64b,66 MHz PCI	16*2*64b,66 MHz PCI	16*2*64b,66 MHz PCI
I/O interconnect B/W	9600 MB/s	9600 MB/s	9600 MB/s

20

Case Study: Future IDISK Servers

IDISK Characteristic	1 TB System	3 TB System	10 TB System
Processors	4 * 1000 MHz	4 * 1000 MHz	4 * 1000 MHz
Memory capacity	8 GB	8 GB	8 GB
SMP Interconnect B/W	30,000 MB/s	30,000 MB/s	30,000 MB/s
Disk capacity	200 * 36 GB	600 * 36 GB	1800 * 36 GB
IDISK Memcpy B/W	200*5,000 MB/s	600*5,000 MB/s	1800*5,000 MB/s
IDISK interconnect B/W	200*2,000 MB/s	600*2,000 MB/s	1800*2,000 MB/s
Disk transfer rate	29 MB/s	29 MB/s	29 MB/s
Disk processor speed	500 MHz	500 MHz	500 MHz
Disk memory	32 MB	32 MB	32 MB
IDISK serial lines	8*2Gbit/s	8*2Gbit/s	8*2Gbit/s

21

Case Study for IDISK: TPC-D Query 1

- Scan 95% - 97% of largest table (“lineitem”) and compute aggregates
- Tremendous reduction in data movement

Scale Factor	1 TB	3 TB	10 TB
Lineitem cardinality (rows)	~6 bill.	~18 bill.	~60 bill.
Lineitem size/Total data moved in SMP (GB)	~870	~2610	~8700
Total data moved in IDISK (KB)	~59	~176	~527

22

Case Study for IDISK: TPC-D Query 1

Inst. per tuple	**200**	**500**	**1000**	**1500**	**2000**	**3000**	**6000**
1 TB							
SMP	157 (s)	157	157	215	285	426	849
IDISK	144	144	144	144	159	253	505
IDISK speedup	1.1x	1.1x	1.1x	1.5x	1.7x	1.7x	1.7x
3 TB							
SMP	293	293	418	620	823	1228	2443
IDISK	144	144	144	144	169	253	505
IDISK speedup	2.0x	2.0x	2.9x	4.3x	4.9x	4.9	4.8x
10 TB							
SMP	922	922	1329	1987	2645	3961	7910
IDISK	160	160	160	160	188	281	562
IDISK speedup	5.8x	5.8x	8.3x	12.4x	14.1x	14.1x	14.1x

*Table shows seconds to scan and process lineitem table
*Speedups from embarrassingly parallel nature of task
*IDISK processing scales better than SMP processing

23

Ongoing Research & Open Questions (I)

- How does IDISK server performance and price compare to
  - Cluster-based shared nothing server? (e.g., NCR WorldMark)
  - CC-NUMA architecture? (e.g., SGI Origin, Sequent NUMA-Q)
- What is right ratio between embedded processors, memory and disks?
- What is performance of large-scale (600-1000 nodes) serial line interconnect?

24

Ongoing Research & Open Questions (II)

- How much processing can we push down into disk?
  - Scans and certain object manipulations are obvious wins
  - What about sort, join, and aggregation operations?
- Does Amdahl's Law limit IDISK performance gains?
  - Exactly how much time is spent doing operations that could be pushed to disk?
  - Answer question by profiling commercial database and doing "atomic benchmarking"
- What's the right programming model? How to safely download code into disk?
- How do we get commercial databases to modularize code so that operations *can* be downloaded to disk processor?

25

Conclusions

- Decision support databases
  - Increasingly important workload
  - Storage and related computation requirements growing faster than processor speed increases
  - Central server processors saturated: current system bottleneck
- IDISK offers architectural alternative
  - Push processing to disk, rather than bringing data to CPU
  - Allows processing of system to scale with increasing storage demand
  - Overcomes pitfalls of previous research attempts
- IDISK advantages
  - Improved performance from exploiting data parallelism
  - Incredible reduction in data movement
  - Reduced cost: trade expensive MIPs for cheap MIPs
- Evolutionary path to completely decentralized system

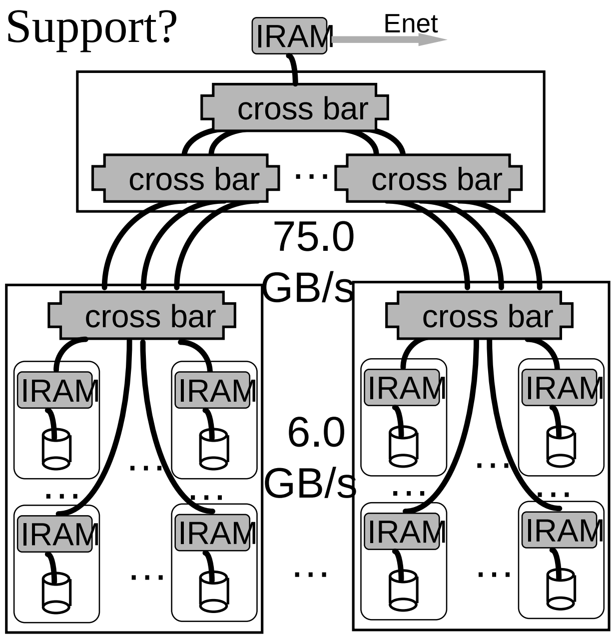
26

Backup Slides

(These slides used to help answer questions.)

27

Revolutionary IDISK: Scalable Decision Support?



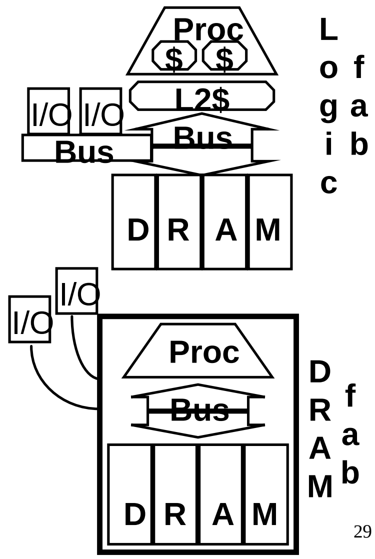
- ✶ 1 IRAM/disk + xbar + fast serial link v. conventional SMP
- ✶ Network latency = f(SW overhead), not link distance
- ✶ Move function to data v. data to CPU (scan, sort, join,...)
- ✶ Looks like cluster ("shared nothing")
- ✶ Cheaper, faster, more scalable (~1/3 $, 3X perf)

28

Berkeley IRAM Vision Statement

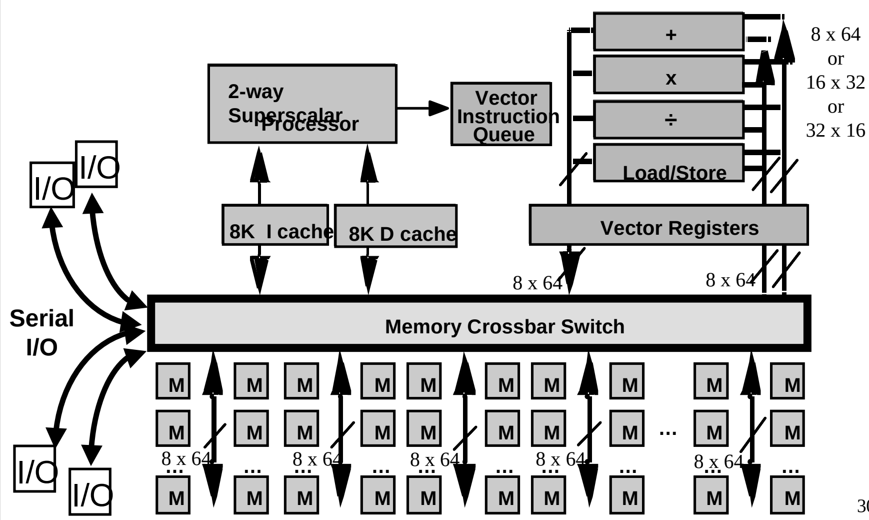
Microprocessor & DRAM on a single chip:

- on-chip memory latency 5-10X, bandwidth 50-100X
- improve energy efficiency 2X-4X (no off-chip bus)
- serial I/O 5-10X v. buses
- smaller board area/volume
- adjustable memory size/width



29

V-IRAM-2: 0.13 µm, Fast Logic, 1GHz 16 GFLOPS(64b)/64 GOPS(16b)/128MB



30

V-IRAM Benefits Database Operations

- Vectorized radix sort (Zagha and Blelloch, Supercomputing '91)
- Vectorized hash join (Rich Martin, UCB)
- Data mining
  - Statistical operations looking for trends in data
- Image/video object manipulations
  - Format conversion
  - Compression
  - Query by

31

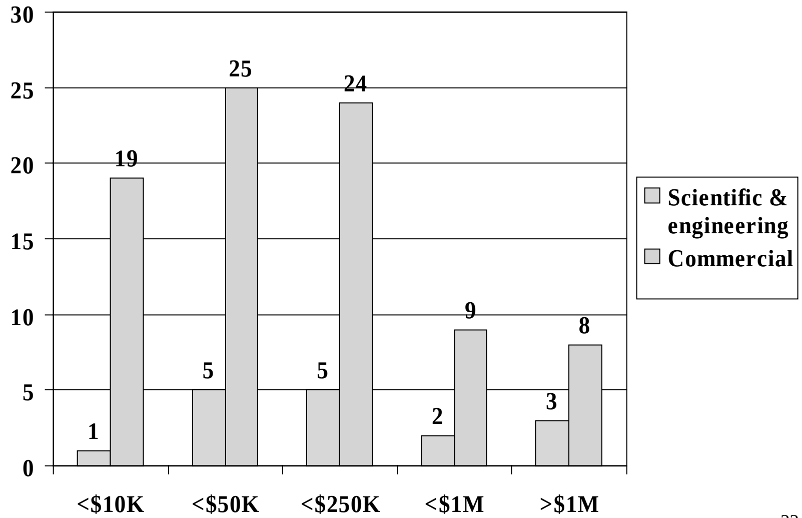
TPC-D Q1

```
SELECT
        L_RETURNFLAG, L_LINESTATUS, SUM(L_QUANTITY) AS
          SUM_QTY,
        SUM(L_EXTENDEDPRICE) AS SUM_BASE_PRICE,
        SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS
          SUM_DISC_PRICE,
        SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)*(1+L_TAX)) AS
          SUM_CHARGE,
        AVG(L_QUANTITY) AS AVG_QTY, AVG(L_EXTENDEDPRICE)
          AS AVG_PRICE,
        AVG(L_DISCOUNT) AS AVG_DISC, COUNT(*) AS
          COUNT_ORDER
    FROM LINEITEM
    WHERE L_SHIPDATE <= DATE '12/1/98' - INTERVAL 'delta' DAYS
    GROUP BY L_RETURNFLAG, L_LINESTATUS
    ORDER BY L_RETURNFLAG, L_LINESTATUS;
```

32

1995 Market Volume by Machine Price

Source: Stenstrom, et al., *IEEE Computer*, December 1997



33

TPC-D Performance Metrics

- Power (QppD)
  - single user query processing power
  - (1 * 3600 * SF) / geometric_mean(Q1…Q17,UF1,UF2)
  - SF = scale factor (e.g., 1, 10, 30, 100, 300, 1000, 3000 GB)
- Throughput (QthD)
  - multi-user query throughput
  - (S * 17 * 3600 * SF) / Ts
  - S = # concurrent users, each executing all 17 read-only queries
  - Ts = total elapsed time
- Queries per hour (QPhd)
  - square_root(QppD * QthD)

34