How Fast Indexing Makes Databases Greener

Michael A. Bender
Stony Brook and Tokutek

Martin Farach-Colton
Rutgers and Tokutek

Bradley C. Kuszmaul
MIT and Tokutek

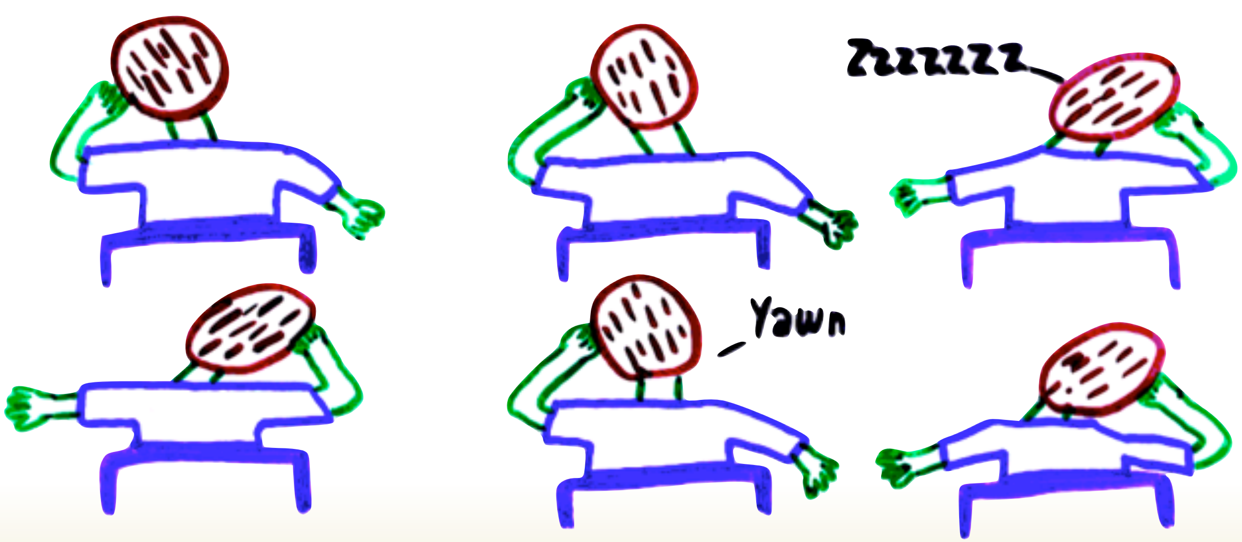
Fast Indexing **Makes Databases Greener**

- Data centers used 1.5% of US electricity in 2006.
- Servers: 50% data-center power
- Storage systems: 27% data-center power

[Battles, Belleville, Grabau, Maurier.'07]

Obligatory reference to EPA study.

Databases are both storage and CPU intensive.

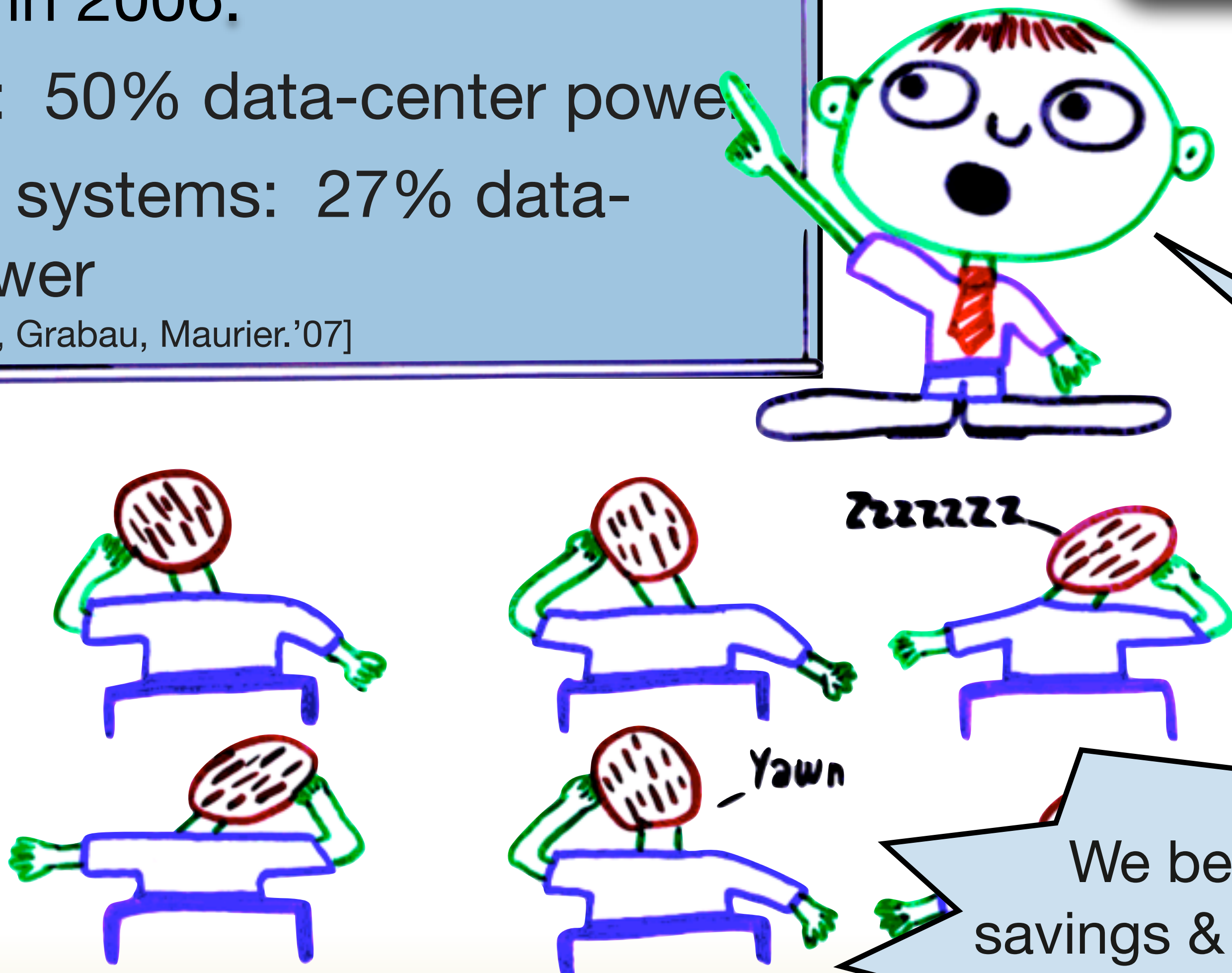


Fast Indexing **Makes Databases Greener**

Obligatory reference to EPA study.

- Data centers used 1.5% of US electricity in 2006.
- Servers: 50% data-center power
- Storage systems: 27% data-center power

[Battles, Belleville, Grabau, Maurier.'07]



Databases are both storage and CPU intensive.

We believe big energy savings & performance gains are still on the table

Modern indexing structures overcome disk-seek bottlenecks of traditional structures

	B-tree	Fractal Tree[R] structure
Insert/delete	$O(\log_B N) = O\left(\frac{\log N}{\log B}\right)$	$O\left(\frac{\log N}{B^{1-\varepsilon}}\right)$

Fast Indexing Makes Databases Greener

Modern indexing structures overcome disk-seek bottlenecks of traditional structures

	B-tree	Fractal Tree[R] structure
Insert/delete	$O(\log_B N)=O\left(\frac{\log N}{\log B}\right)$	$O\left(\frac{\log N}{B^{1-\varepsilon}}\right)$

- If B=1024, then $B/\log B \approx 100$. → *100x* speedup.
  (No asymptotic loss in point queries.)

STONY BROOK Rutgers MIT Tokutek

Fast Indexing Makes Databases Greener

Modern indexing structures overcome disk-seek bottlenecks of traditional structures

	B-tree	Fractal Tree[R] structure
Insert/delete	$O(\log_B N) = O\left(\frac{\log N}{\log B}\right)$	$O\left(\frac{\log N}{B^{1-\varepsilon}}\right)$

- If B=1024, then $B/\log B \approx 100$. → *100x* speedup.
  (No asymptotic loss in point queries.)

- Other structures supporting fast inserts:

[O'Neil[1],Cheng[2], Gawlick[3], O'Neil 96] [Argel 03] [Graefe 03] [Brodal, Fagerberg 03]

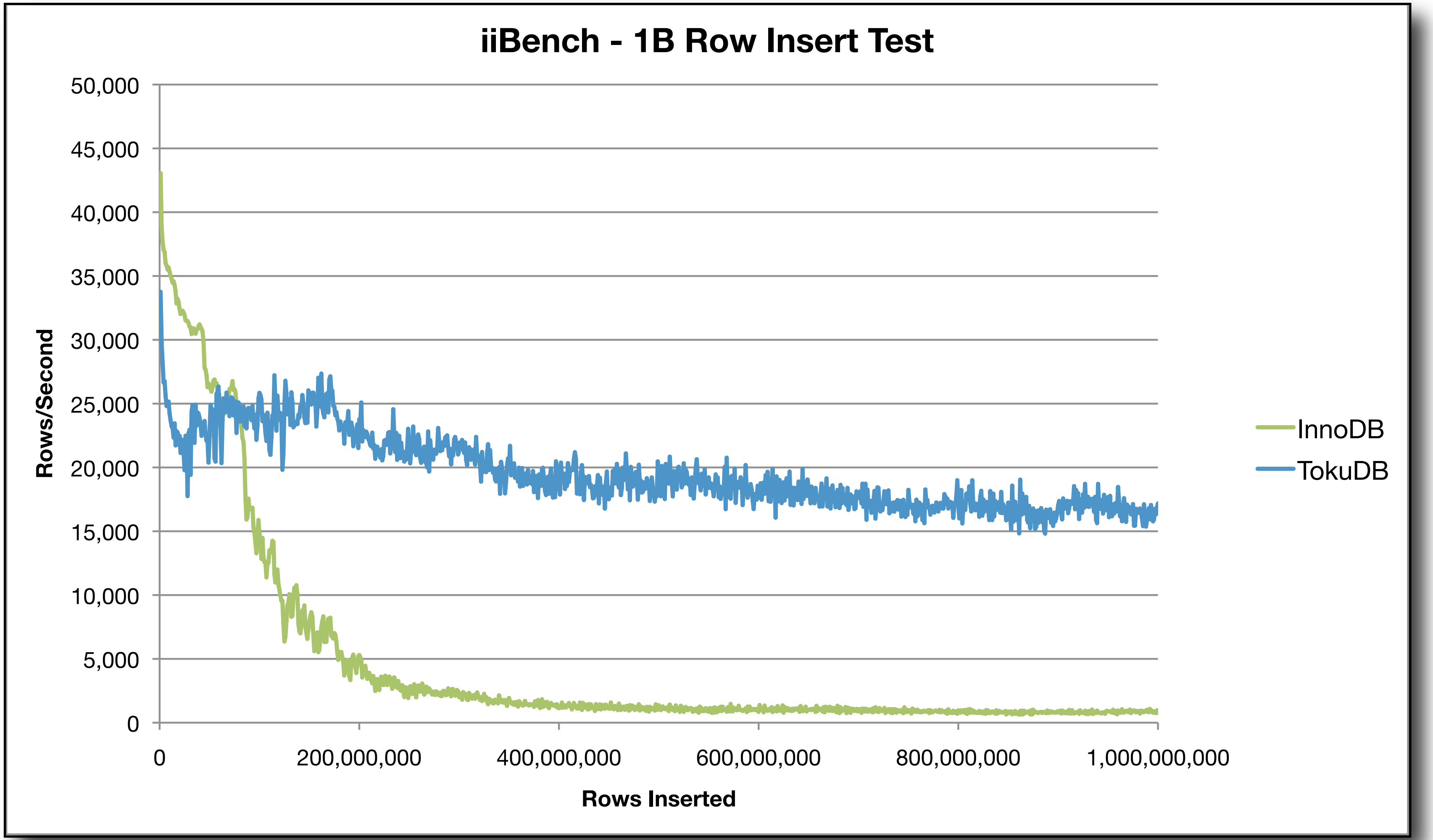
[Buchsbaum, Goldwasser, Venkatasubramanian, Westbrook 00]

[Brodal, Demaine, Fineman, Iacono, Langerman, Munro 00]

Fast Indexing Makes Databases Greener

Ex. TokuDB^R supports >20,000 index inserts/sec even on high-entropy workloads.

- Effectively transform random I/O into sequential I/O.

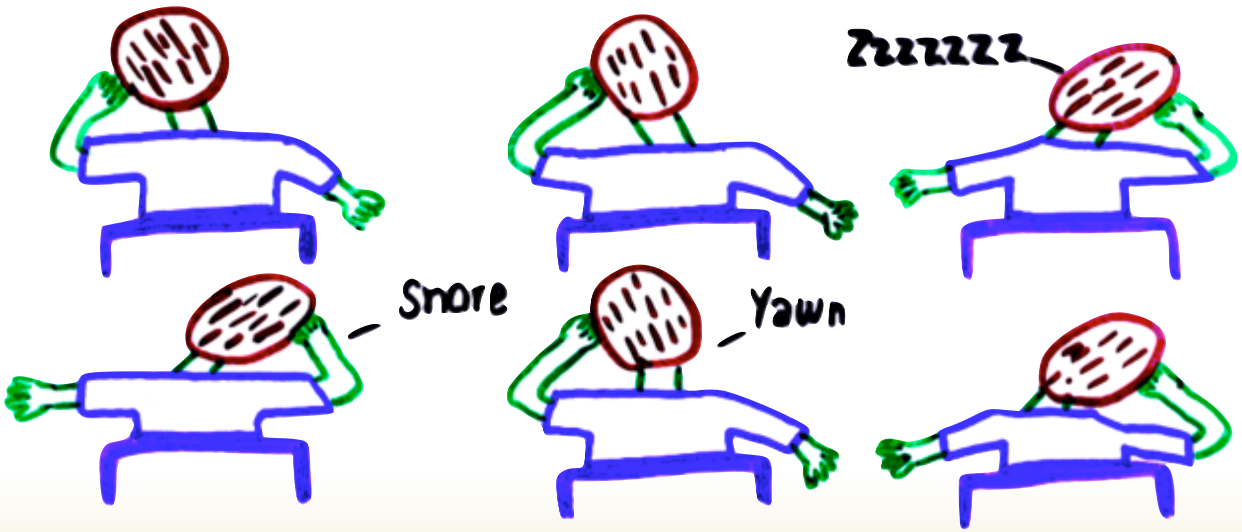


one reason why ^ Fast Indexing Makes Databases Greener

Fast insertions means

➡ we can efficiently maintain sophisticated indexes,

➡ both insert & query-dominated workloads also can be more energy-efficient.

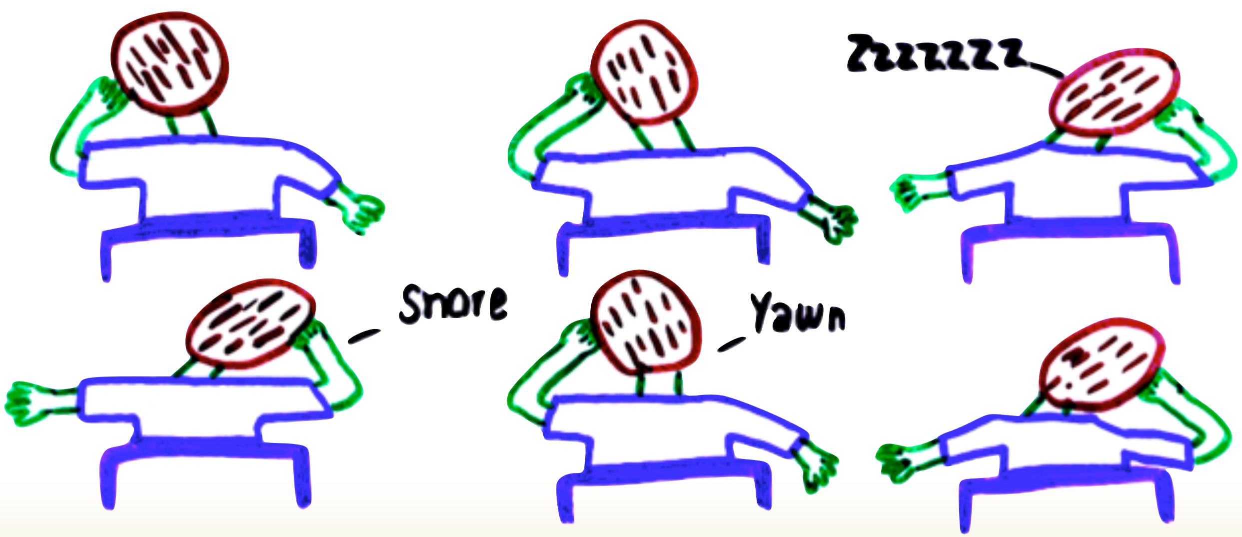


one reason why Fast Indexing Makes Databases Greener

Fast insertions means

➡ we can efficiently maintain sophisticated indexes,

➡ both insert & query-dominated workloads also can be more energy-efficient.



one reason why ^ Fast Indexing Makes … ses Greener

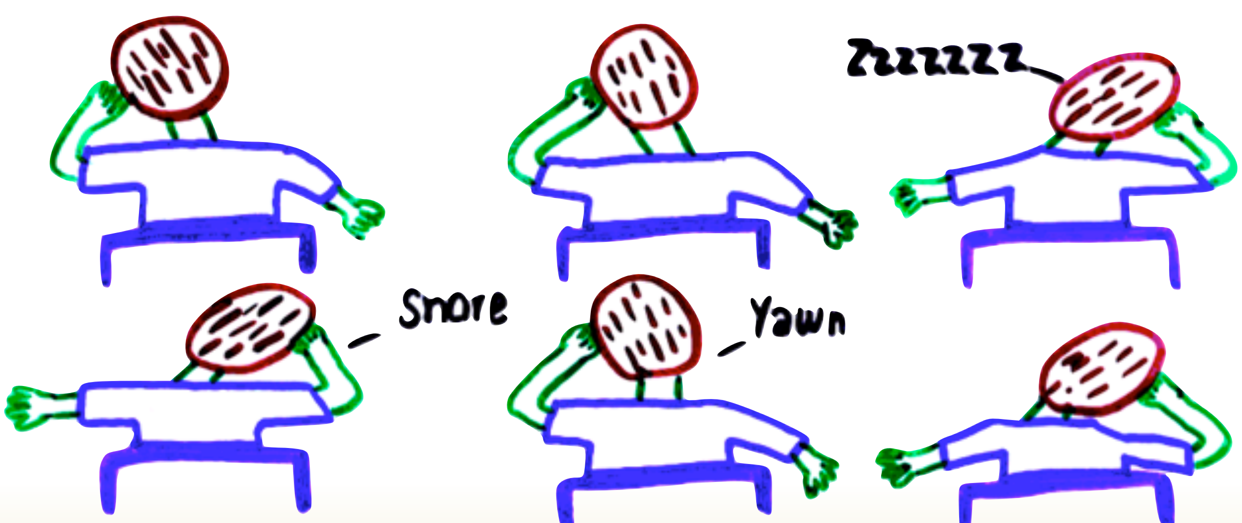
Fast insertions means

➡ we can efficiently maintain sophisticated indexes,

➡ both insert & query-dominated workloads also can be more energy-efficient.

customer hat

Many users who *think* they have query bottlenecks actually have *insertion* bottlenecks. Customer issues can be solved by fast inserts into sophisticated indexes.



another reason why ^ Fast Indexing Makes Databases Greener

Promise of green algorithms:
enable more power-efficient hardware.

Data centers are already designed around algorithmic specs because existing algorithms should run well on existing hardware.

Algorithms + Enabled Hardware = Big Win

another reason why ^ Fast Indexing Makes Databases Greener

Example: Data centers use many small-capacity disks rather than a few large-capacity disks

- Why? One reason is to get more I/Os.
- *Fractal Tree indexes don't need more spindles.*

Power consumption of disks

- Enterprise 80 to 160 GB disk runs at 4W (idle power).
- Enterprise 1-2 TB disk runs at 8W (idle power).

another reason why ^ Fast Indexing Makes Databases Greener

Example: Data centers use many small-capacity disks rather than a few large-capacity disks

- Why? One reason is to get more I/Os.
- *Fractal Tree indexes don't need more spindles.*

Power consumption of disks

- Enterprise 80 to 160 GB disk runs at 4W (idle power).
- Enterprise 1-2 TB disk runs at 8W (idle power).

Savings on the table: ~10x in storage

- Other considerations modify this factor
  - e.g., CPUs necessary to drive disks, scale-out infrastructure, cooling, etc.

Algorithms + Enabled Hardware = Big Win

Open Prob 1: Highly Concurrent & Multithreaded Indexing

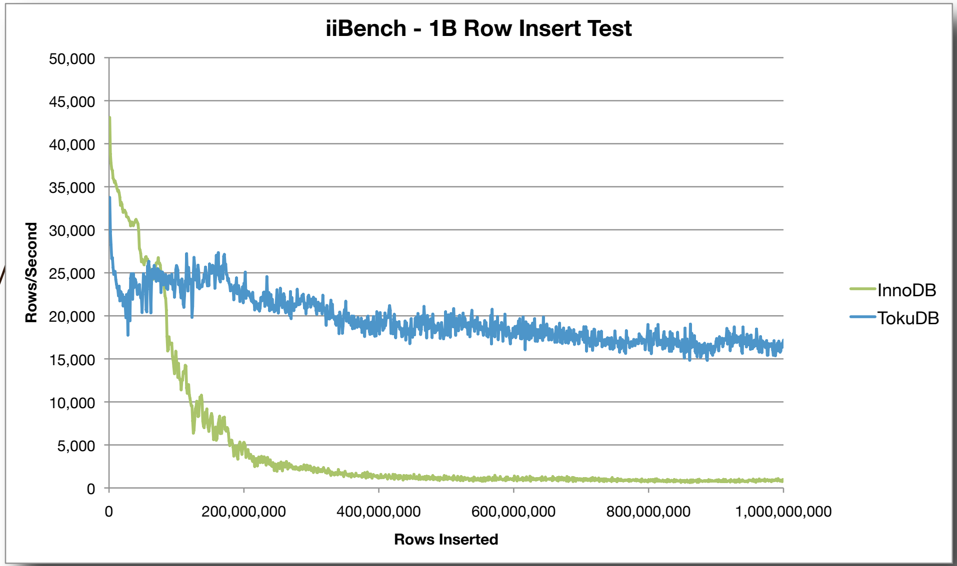
Develop concurrent, multithreaded indexing data structures for slow, high-core-count machines

- server CPU: ~100 W
- laptop CPU: 5-10 W
  - 4x less capable, 10-20x less power hungry
  - **5x more energy efficient**
- mobile-phone CPU
  - **another factor of 5 is on the table**

Open Prob 1: Highly Concurrent & Multithreaded Indexing

Develop concurrent, multithreaded indexing data structures for slow, high-core-count machines

- server CPU: ~100 W
- laptop CPU: 5-10 W
  - 4x less capable, 10-20x less power hungry
  - **5x more energy efficient**
- mobile-phone CPU
  - **another factor of 5 is on the table**



Fractal Trees drive more CPUs than B-trees

- CPU intensive. e.g, TokuDB is CPU bound
- big efficiency gains are on the table

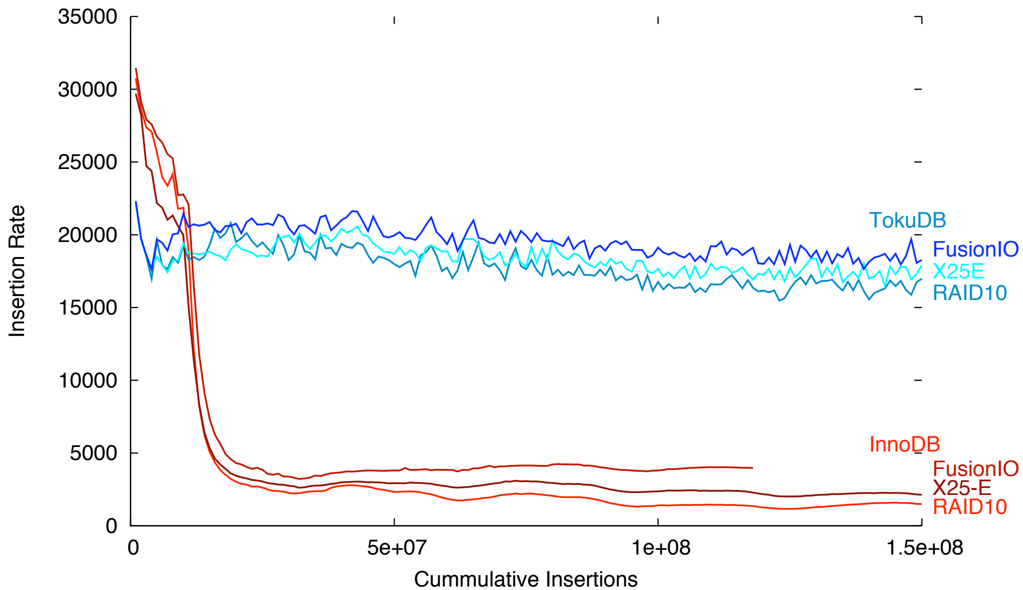
STONY BROOK RUTGERS MIT Tokutek

Open Prob 2: Energy-Efficient SSD/Rotational Disk Hybrid

Design a SSD/rotational disk hybrid for a Fractal-Tree-based storage system.

- Rotational devices are more efficient for sequential I/O
- SSDs are more efficient for random I/O.

Can a hybrid offer energy savings by using each device for the workload it is best suited for?

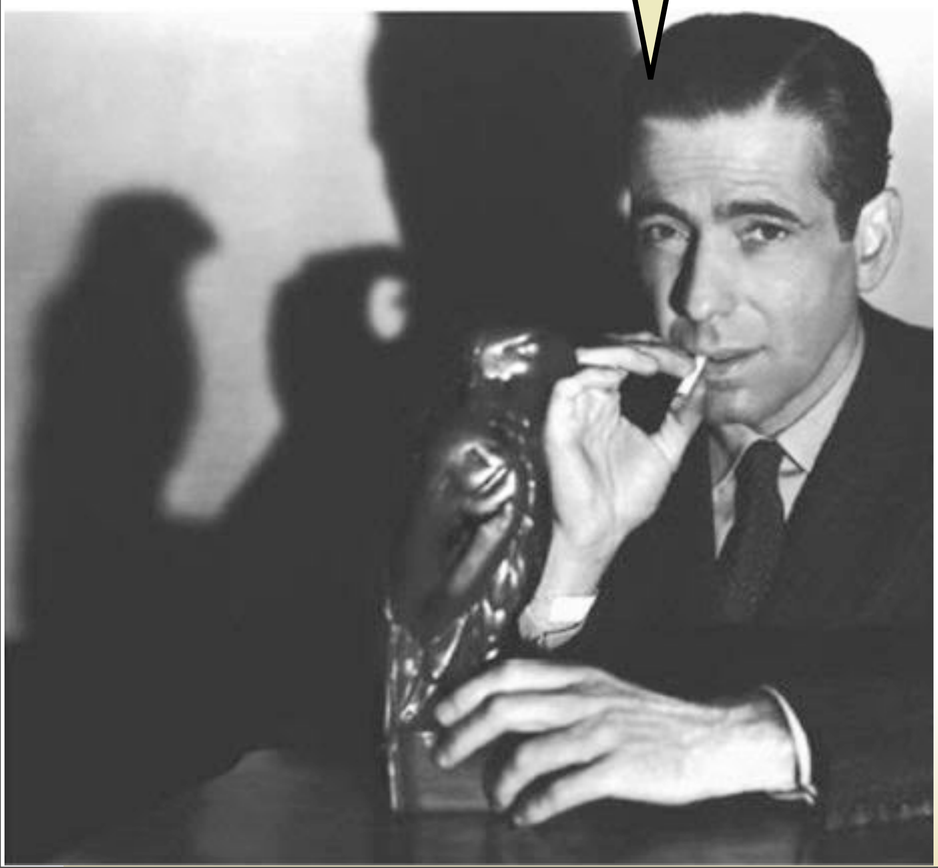


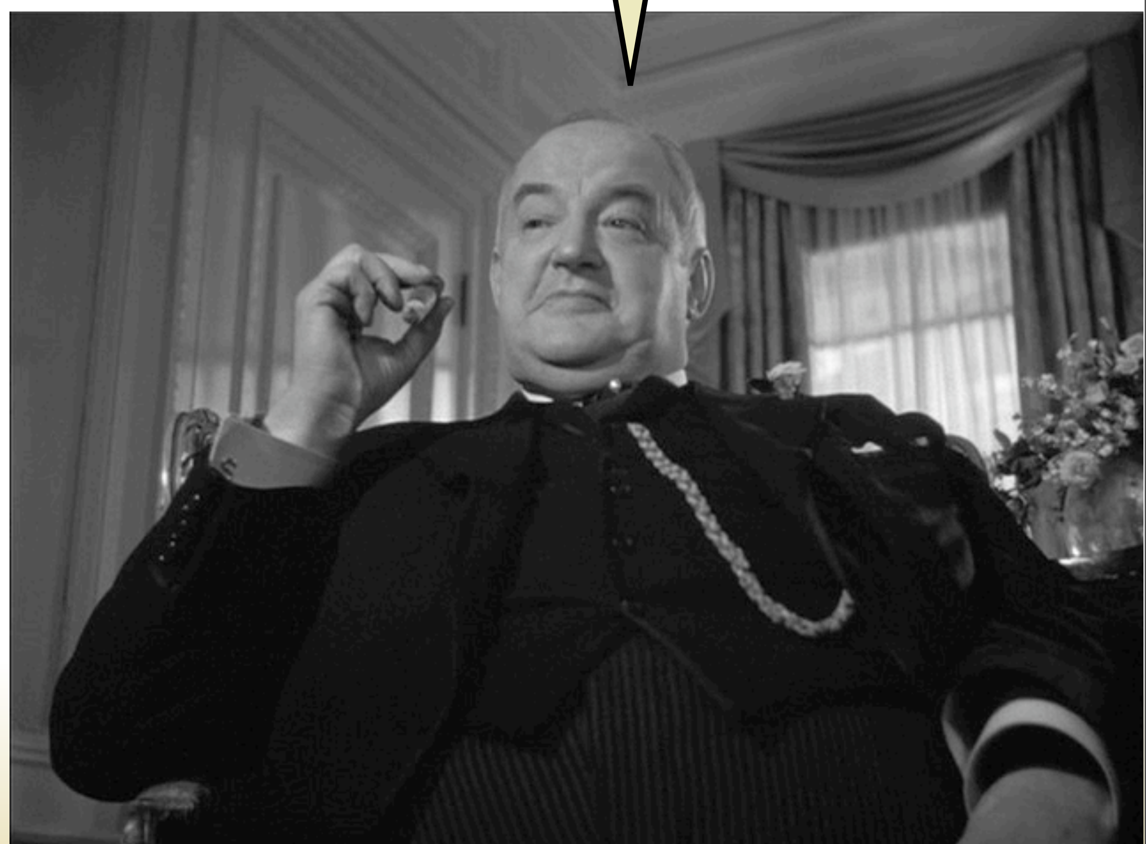
Fractal Trees deliver >10x speedups on SSDs vs B-trees

Open Prob 3: The proof is in the pudding

Ten thousand? We were talking about a lot more money than this.

Yes, sir, we were, but this is genuine coin of the realm. With a dollar of this, you can buy ten dollars of talk.

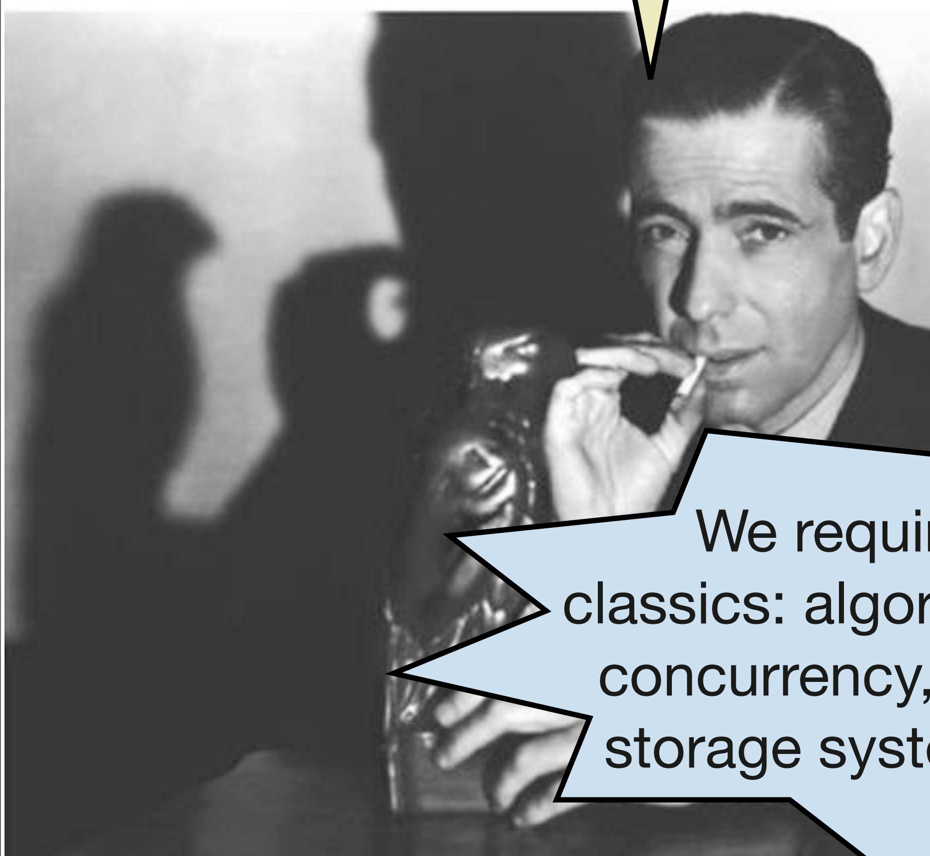


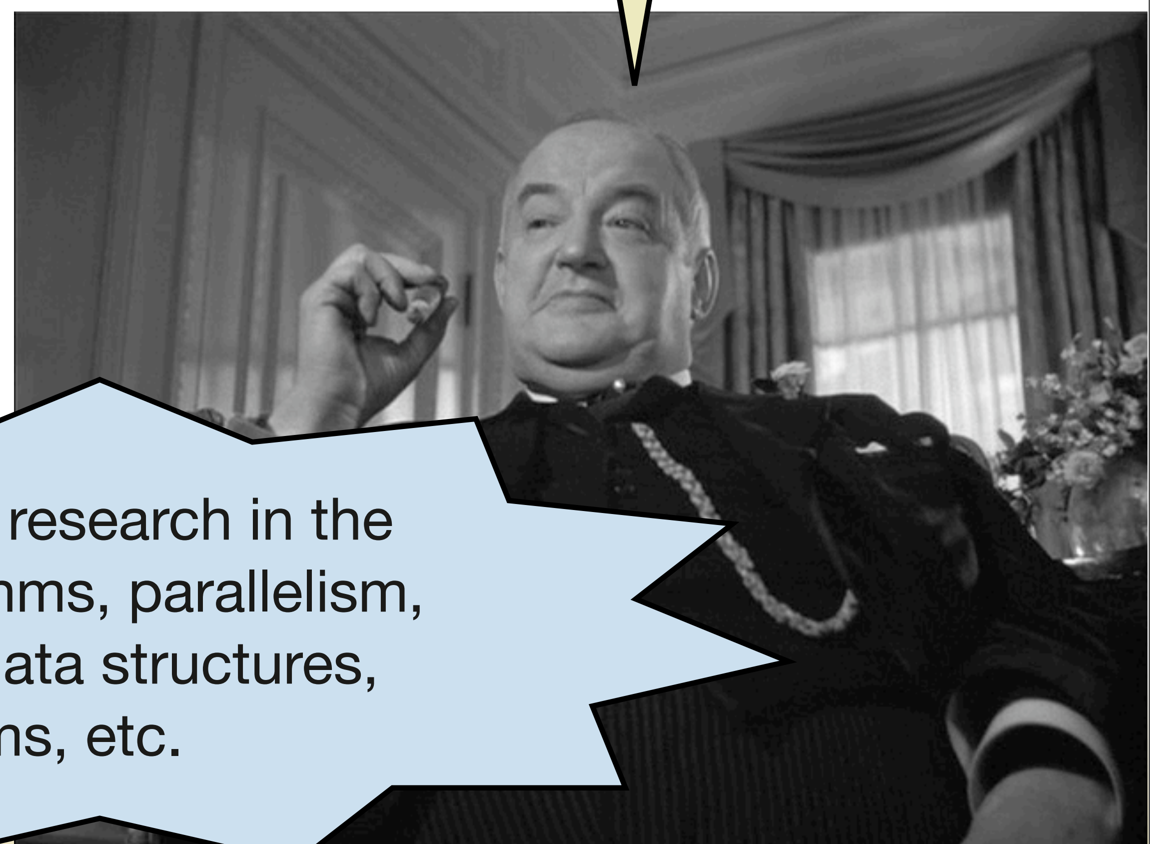


Open Prob 3: The proof is in the pudding

Ten thousand? We were talking about a lot more money than this.

Yes, sir, we were, but this is genuine coin of the realm. With a dollar of this, you can buy ten dollars of talk.





We require research in the classics: algorithms, parallelism, concurrency, data structures, storage systems, etc.