Free Transactions with Rio Vista

David E. Lowell

Peter M. Chen

Rio Project

Electrical Engineering and Computer Science

University of Michigan

David E. Lowell

University of Michigan

The Problem

Writing or modifying permanent data is dangerous

Example: Transferring money

1. **Deduct $1,000,000 from Dave's account**
2. **Add $1,000,000 to Pete's account**
3. **Pete signs Dave's thesis**

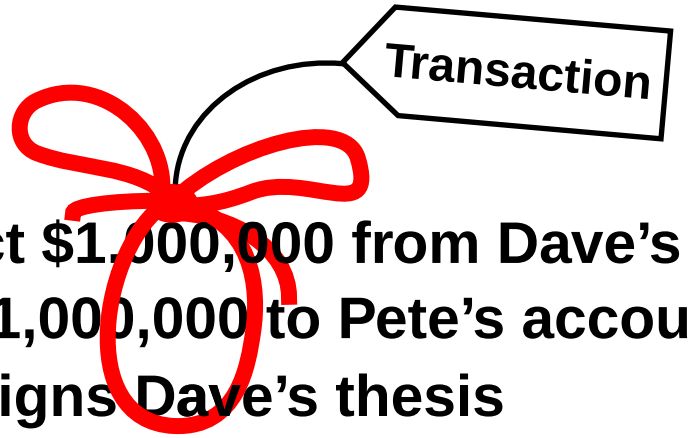
A crash in the middle of these steps leaves things corrupted!

David E. Lowell

University of
Michigan

A Solution

Use transactions to group these steps into an indivisible unit.



1. **Deduct $1,000,000 from Dave's account**
2. **Add $1,000,000 to Pete's account**
3. **Pete signs Dave's thesis**

David E. Lowell

University of Michigan

Transactions on Memory

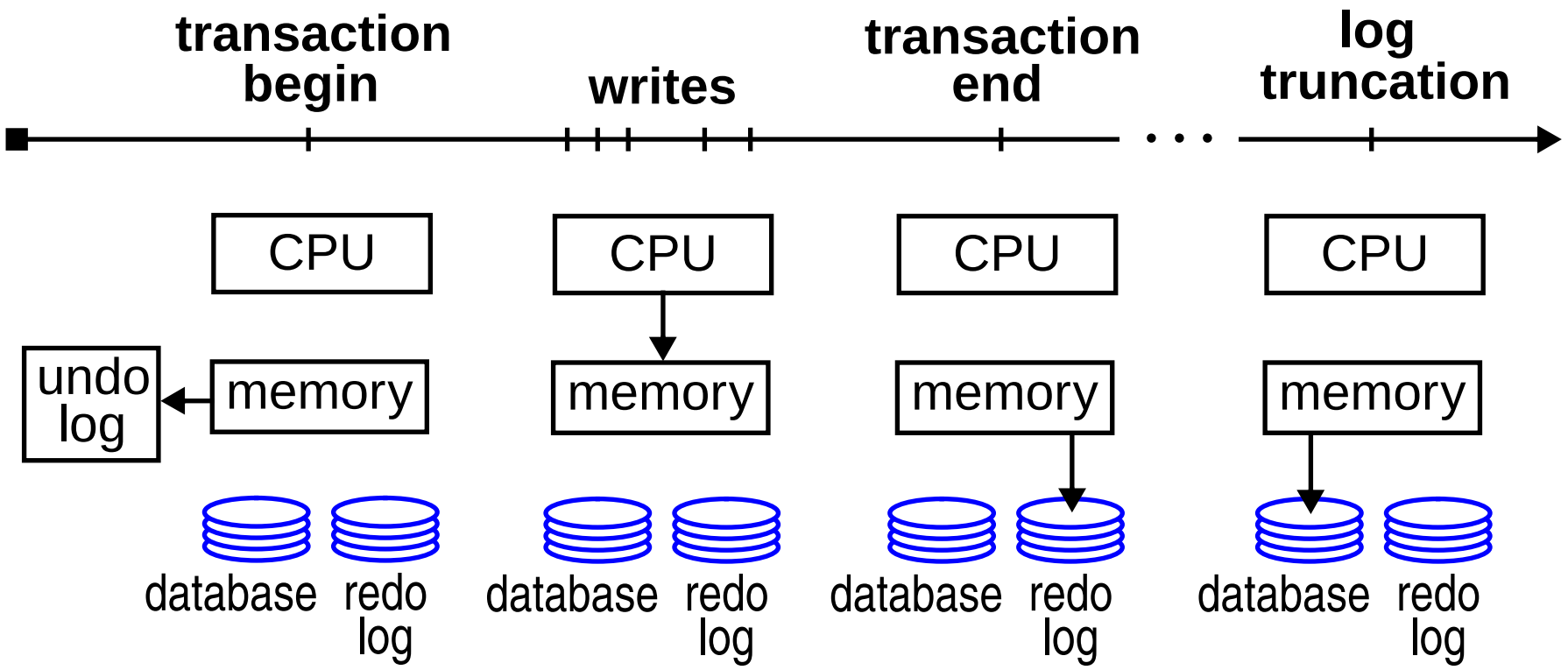
An area of memory made persistent and updated within transactions

Essence of a transaction

- **Just atomicity, durability**
- **No nesting, concurrency control**

RVM is an example

A Transaction on Memory



David E. Lowell

University of
Michigan

But...

Transactions are slow

- **disk writes**
- **system calls**
- **copies**
- **log management**

Rio can help...

David E. Lowell

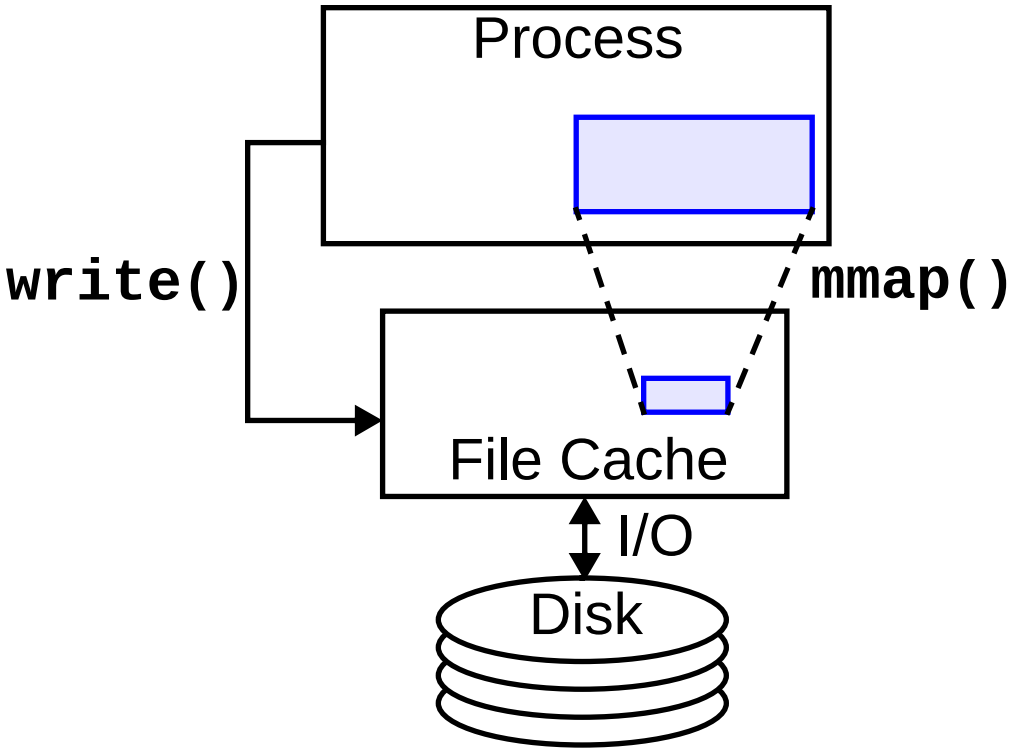
University of
Michigan

Introduction to Rio

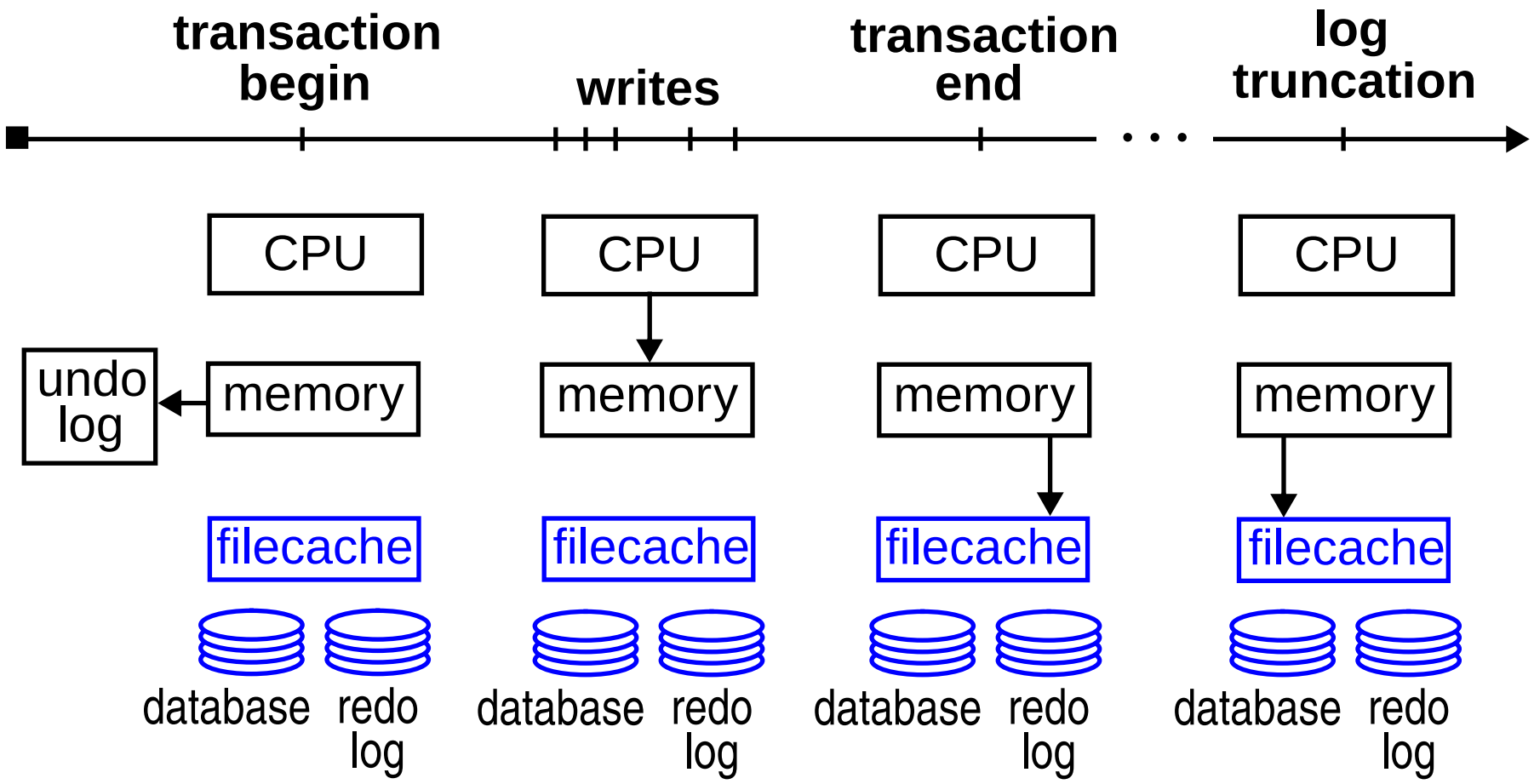
Protects file cache during crash

Restores file cache contents on reboot

→ file cache pages are persistent!



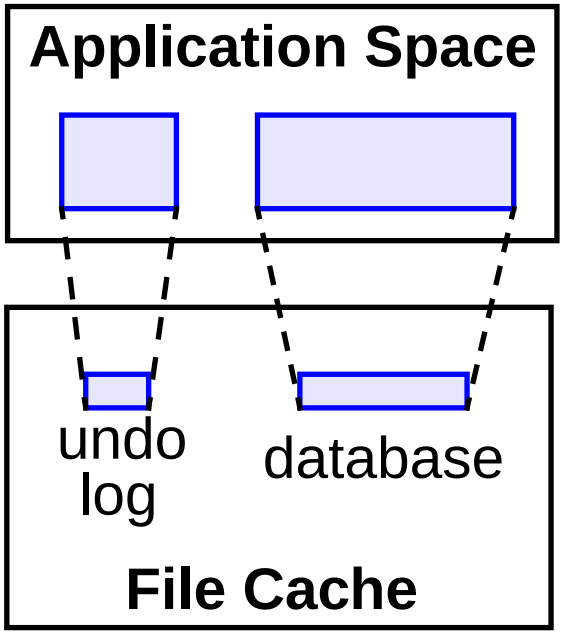
A Transaction with Rio



David E. Lowell

University of
Michigan

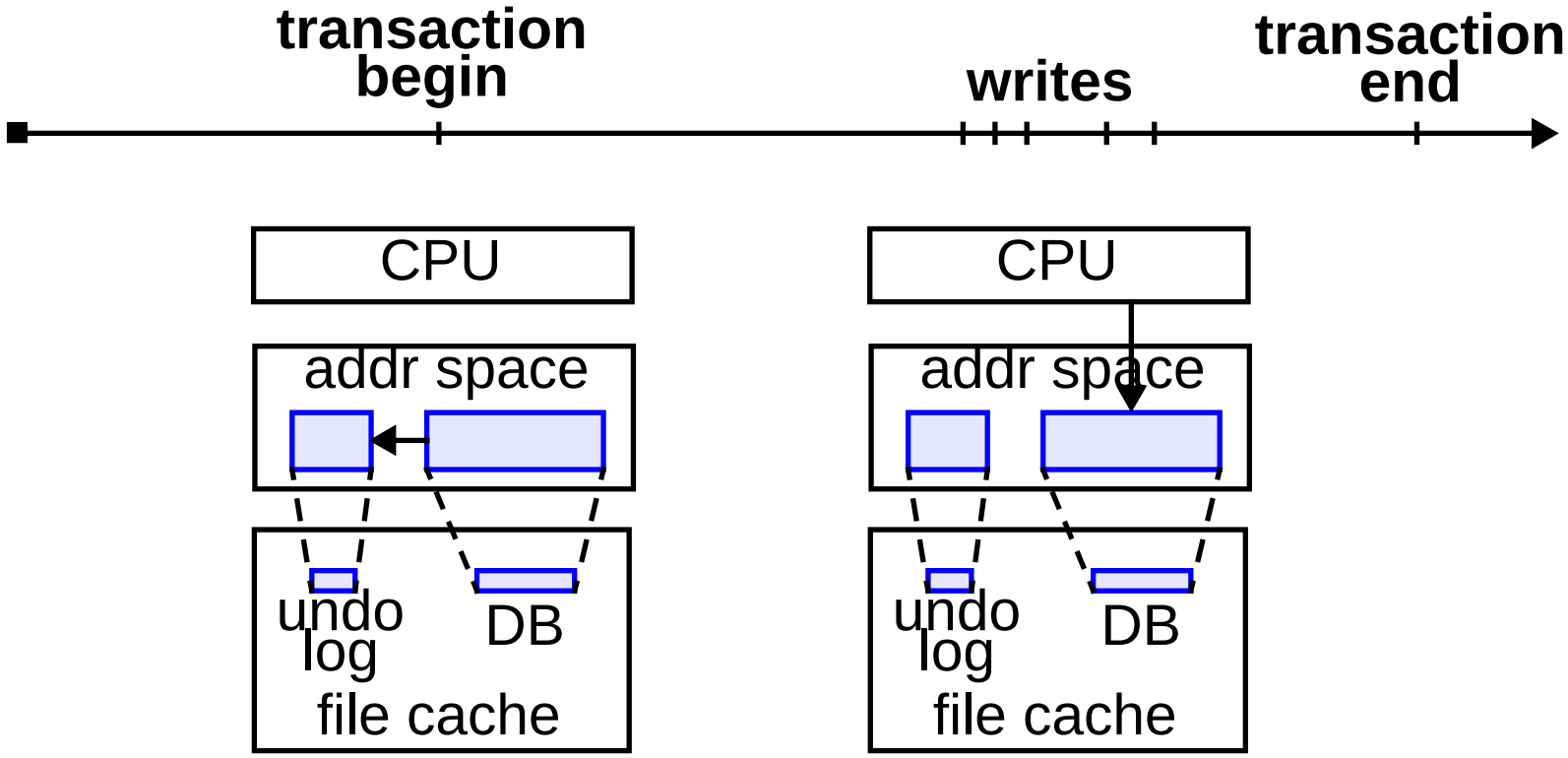
Vista: A Transaction System Tailored for Rio



Maps persistent memory from Rio file cache

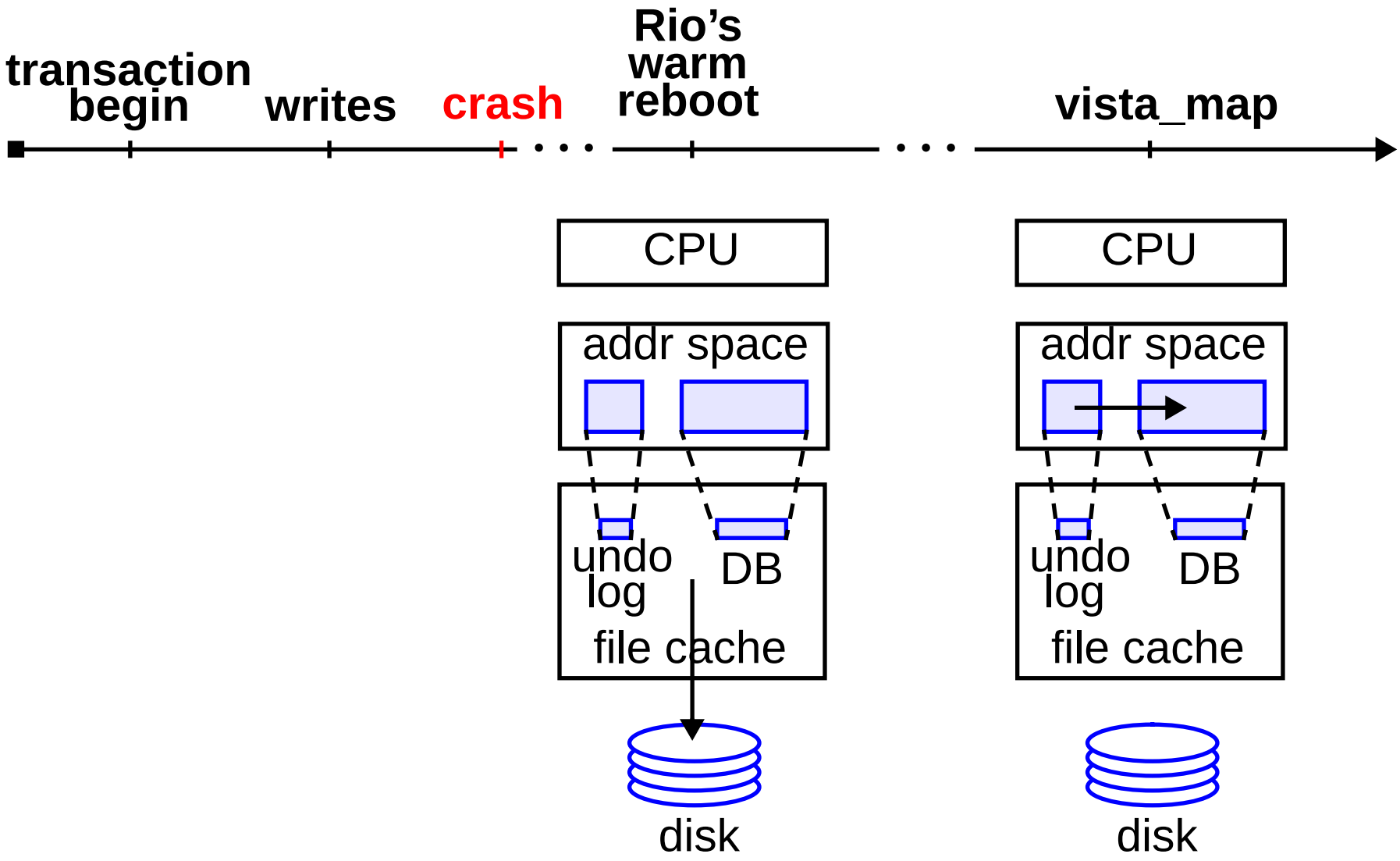
- **all updates immediately permanent**
- **no redo log needed**
- **just undo updates on abort/crash**

A Vista Transaction



1. **Vista copies before-images to undo log**
2. **Application directly writes permanent data**
3. **Vista discards undo log on commit**

Recovery in Vista



David E. Lowell

University of
Michigan

Vista Highlights

Simplicity and Performance

- no disk I/O
- no redo log
- simple recovery
- only 1 data copy
- no system calls
- only 720 lines in size
- scales linearly with CPU speed

David E. Lowell

University of
Michigan

Performance Evaluation

Benchmarks:

Synthetic, Debit-Credit, Order-Entry

Systems:

RVM, RVM-Rio, Vista

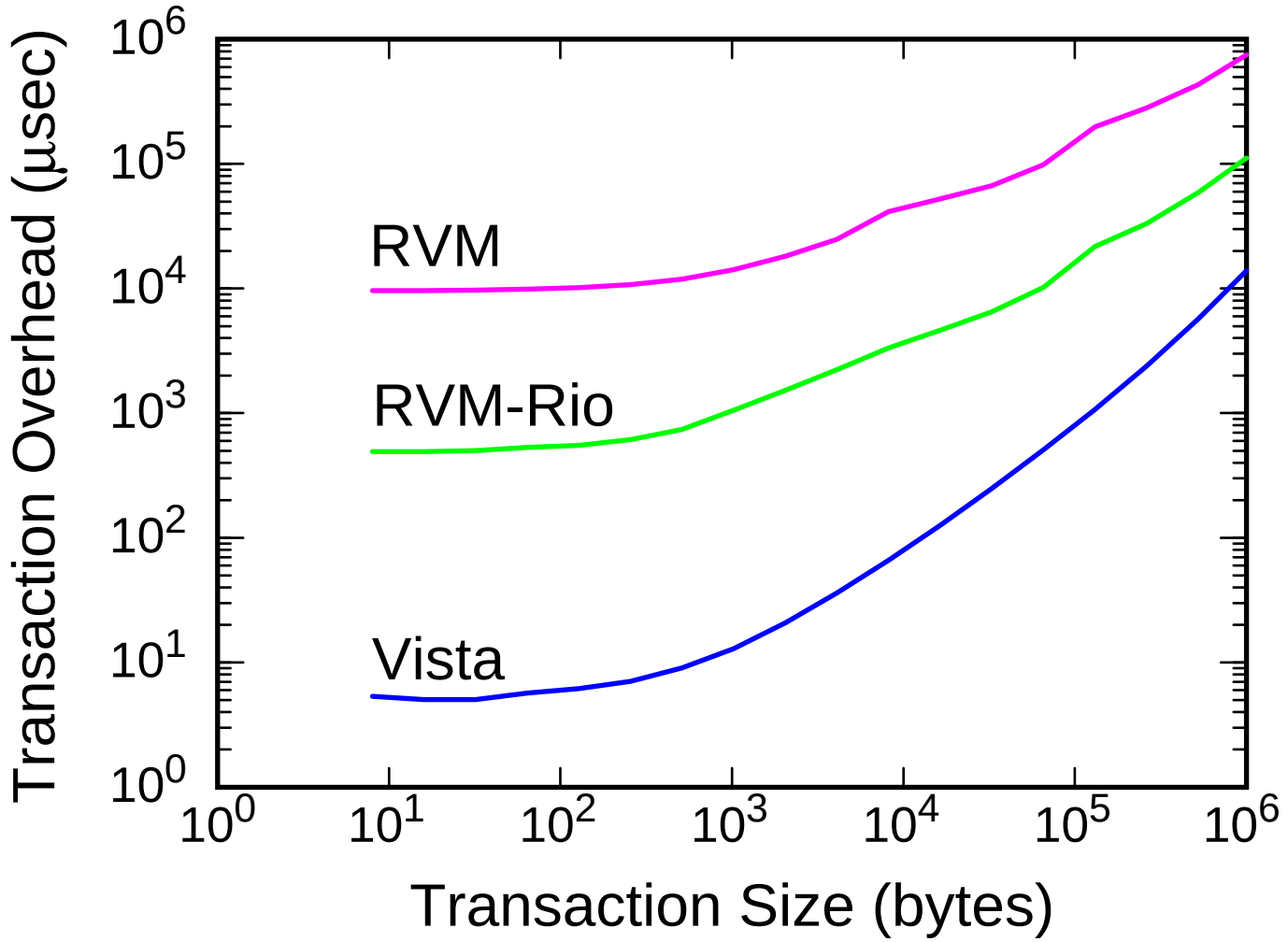
Platform:

175 MHz DEC Alpha 3000/600 workstation with 256 MB of memory. Separate disks for RVM database, log, and swap.

David E. Lowell

University of
Michigan

Transaction Overhead

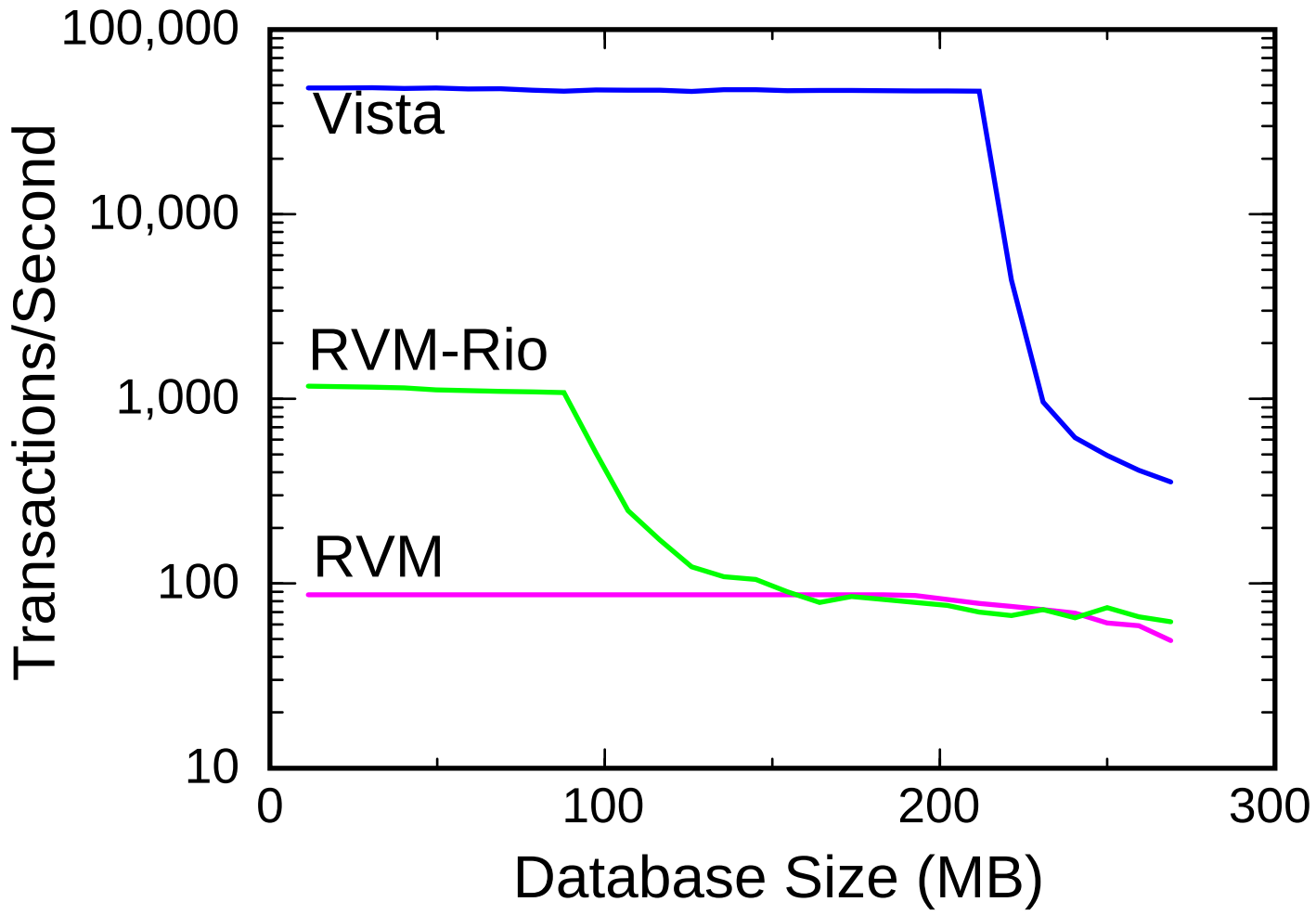


synthetic benchmark

David E. Lowell

University of
Michigan

Transaction Throughput

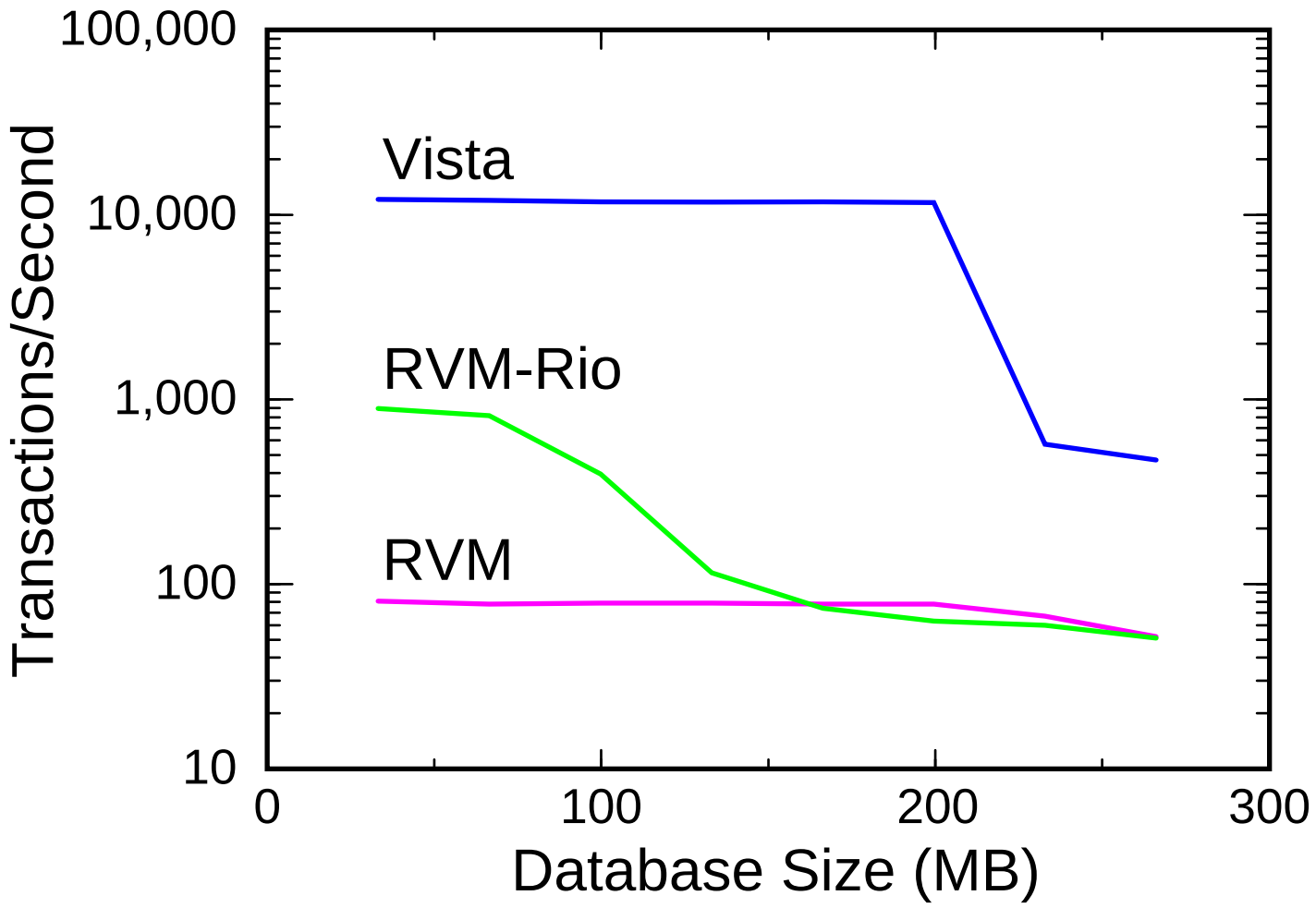


debit-credit benchmark

David E. Lowell

University of
Michigan

Transaction Throughput



order-entry benchmark

David E. Lowell

University of
Michigan

Using Vista

What we do now:

- Persistent VM
- Fine grained transactions
- Explicit or implicit logging
- Do mallocs and frees within transactions

Extending and using free transactions:

- Send and receive messages within transactions
- Build inexpensive checkpointing and persistent processes
- Build a DSM with persistence and transactions

David E. Lowell

University of
Michigan

Conclusions

Vista is really fast

- **Rio speeds up RVM by 20x**
- **Vista gets another 100x (2000x total)**

You can do neat stuff with free transactions

- **Use transactions for fine-grained tasks**
- **Reliable atomic messages, persistent DSM, persistent processes, ...**

`http://www.eecs.umich.edu/Rio`

David E. Lowell

University of Michigan

Isn’t Vista trivial?

RVM-Rio and Vista performance gap is still surprising

Vista’s simplicity is an interesting result

- **It’s really small**
- **It’s really simple**
- **It’s really fast**
- **Isn’t that the ideal?**

David E. Lowell

University of Michigan

What about concurrency?

Vista provides minimal transactions

- **Many applications are single threaded**
- **Concurrency schemes can be added as needed in manner appropriate for each application**
- **Given ultra-fast transactions, concurrency schemes can be simpler**

The lower the overhead of transactions, the more places they'll be useful

David E. Lowell

University of Michigan

Is Vista DB vulnerable?

Only two known quantitative studies of this effect

- **Both Chen96 and Ng97 show data not corrupted substantially more than disk**

We map metadata in a separate region from user data

- **Fixes “off-by-one” and “buffer overrun” errors which are very common failure modes**

Updates within transactions help

Could use sandboxing or similar technique if really necessary

David E. Lowell

University of
Michigan

What about BIG databases?

No database works well when thrashing

- **Vista's reliable memory might help a bit**
- **No double buffering or double paging**

Vista is optimized for working sets that fit

Vista is targeted for new, finer grained applications

David E. Lowell

University of Michigan

What about group commit?

Group commit improves throughput for a group of transactions

Vista improves latency for a single transaction

Group commit is not a general solution

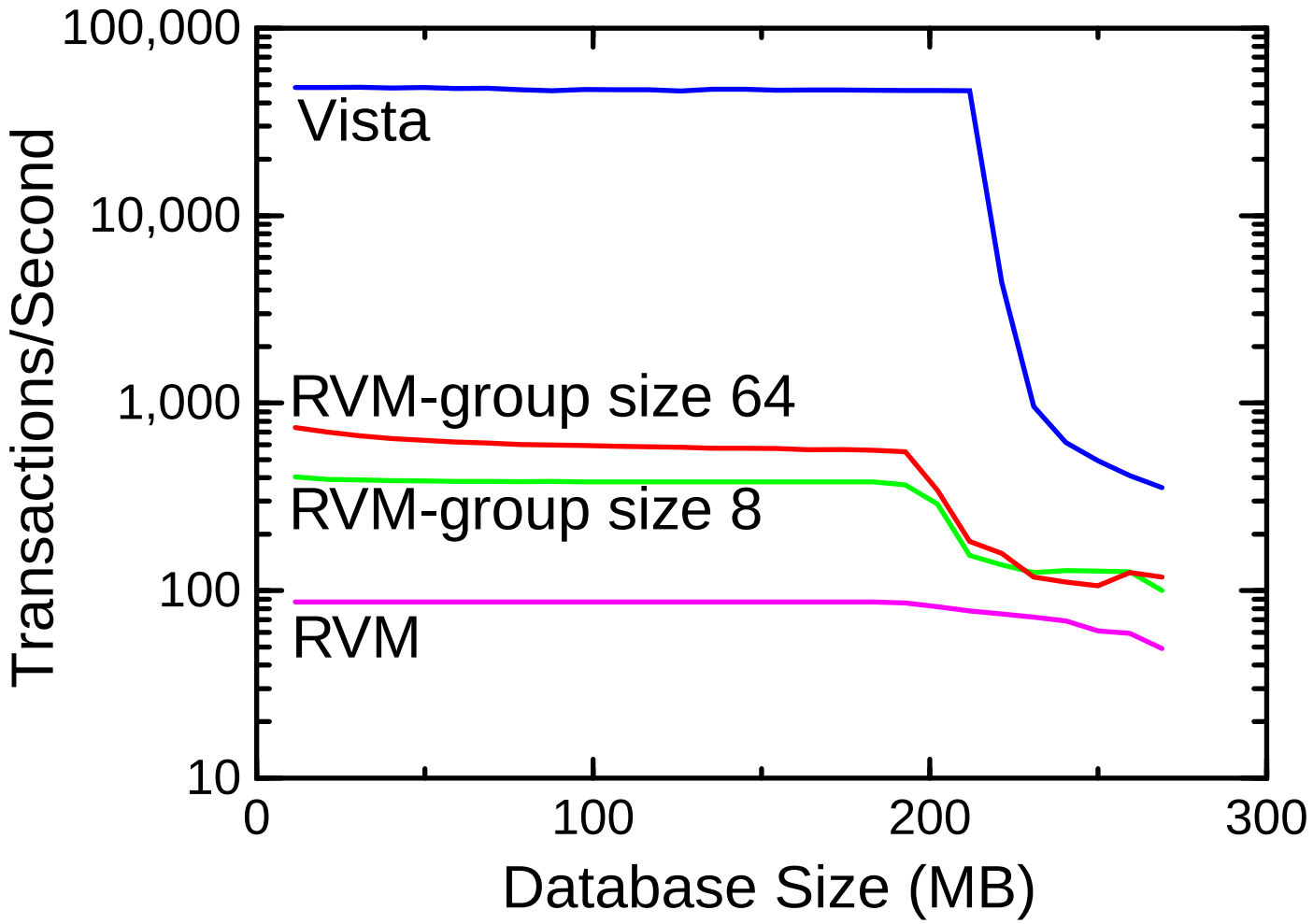
Group commit is limited by disk bandwidth

Vista is still 100x better than group commit

David E. Lowell

University of
Michigan

Transaction Throughput



debit-credit benchmark with group commit