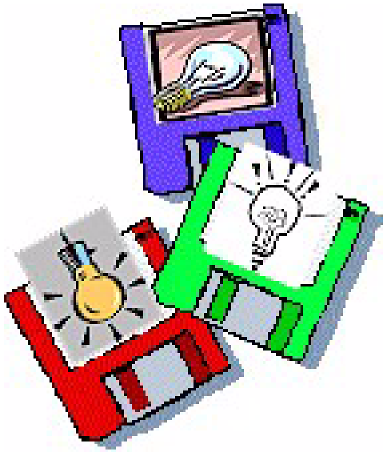


Data Mining on an OLTP System (Nearly) for Free

Erik Riedel
Hewlett-Packard Labs

Greg Ganger, Christos Faloutsos, Dave Nagle
Carnegie Mellon University

Outline

Motivation

Freeblock Scheduling

Scheduling Trade-Offs

Performance Details

Applications

Related Work

Conclusion & Future Work

Disk Trends

	1980	1987	1990	1994	1999	80-99
Model	IBM 3330	Fujitsu M2361A	Seagate ST-41600n	Seagate ST-15150n	Quantum Atlas 10k	Annual Improvement
Average Seek	38.6 ms	16.7 ms	11.5 ms	8.0 ms	5.0 ms	11% / year
Rotational Speed	3,600 rpm	3,600 rpm	5,400 rpm	7,200 rpm	10,000 rpm	6% / year
Capacity	0.09 GB	0.6 GB	1.37 GB	4.29 GB	18.2 GB	32% / year
Bandwidth	0.74 MB/s	2.5 MB/s	3-4.4 MB/s	6-9 MB/s	18-22 MB/s	20% / year
8 KB Transfer	65.2 ms	28.3 ms	18.9 ms	13.1 ms	9.6 ms	11% / year
1 MB Transfer	1,382 ms	425 ms	244 ms	123 ms	62 ms	18% / year

Trends in single drive performance

- **huge capacity increases**
- **bandwidth doesn’t keep pace**
- **seek/rotation lagging far behind**

Outline

Motivation

Freeblock Scheduling

Scheduling Trade-Offs

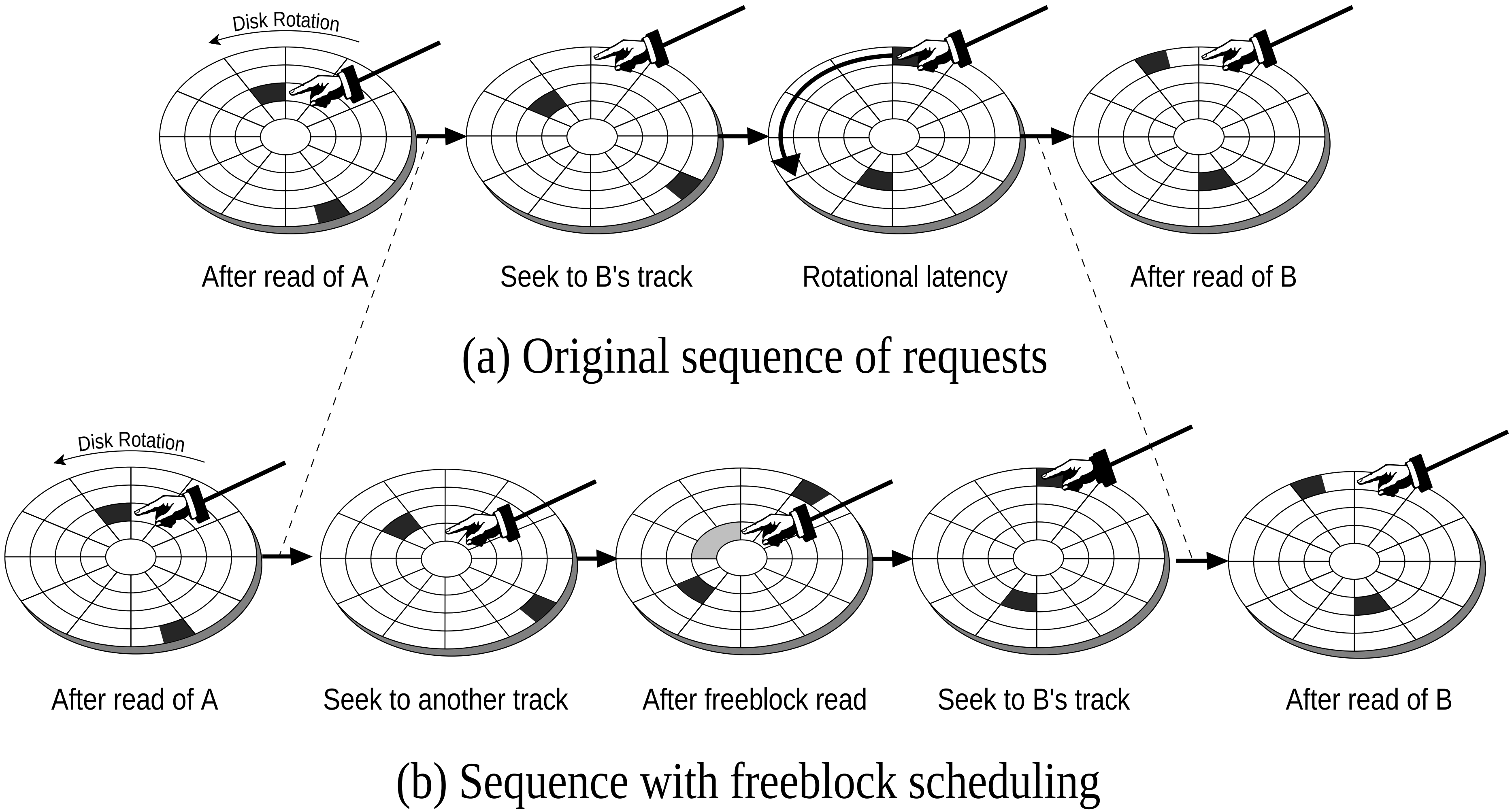
Performance Details

Applications

Related Work

Conclusion & Future Work

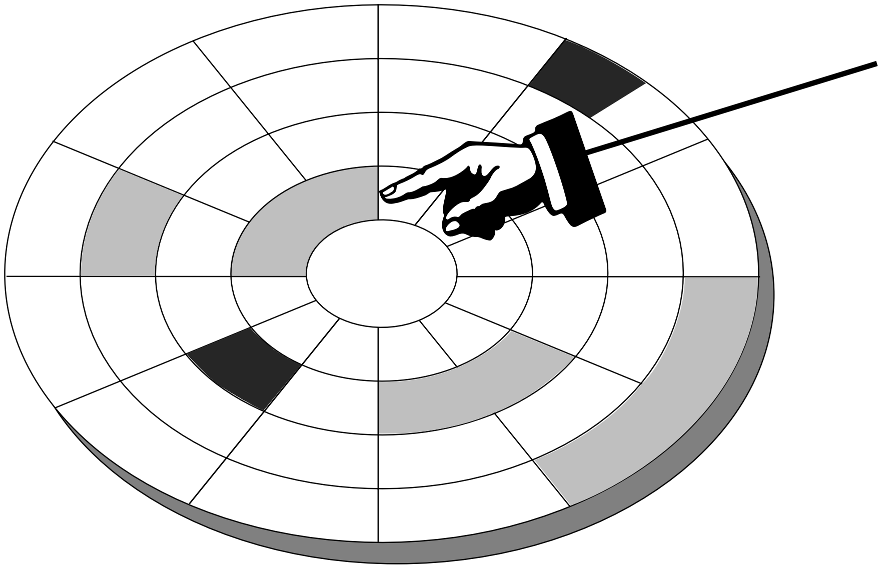
Freeblock Scheduling



Background work during positioning time

- **process sequential workload during “idle” foreground time**

Freeblock Opportunities



ordering of processing blocks does not affect the result

```
foreach block(B) in relation(X)
{
      process(B) -> B'
}
combine(B') -> result(R)
```

Freeblock choices

Most effective background workloads

- scan across a large number of blocks
- order of processing blocks doesn't matter
- "opportunistic" performance acceptable

Outline

Motivation

Freeblock Scheduling

Scheduling Trade-Offs

Performance Details

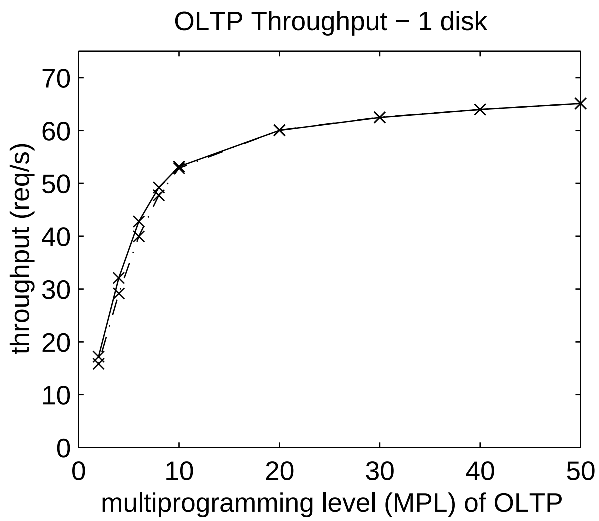
Applications

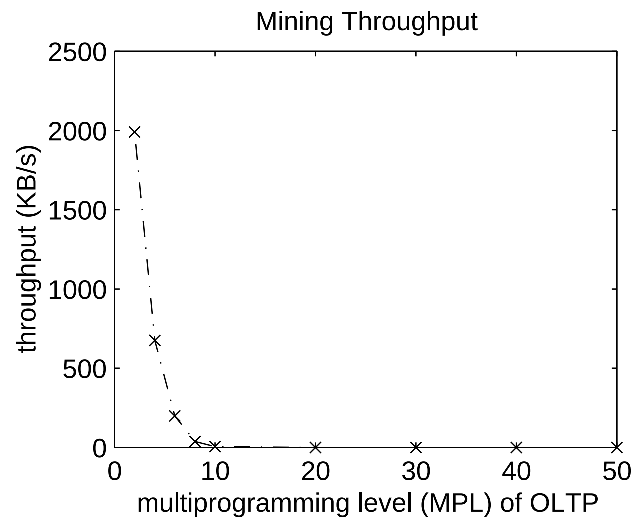
Related Work

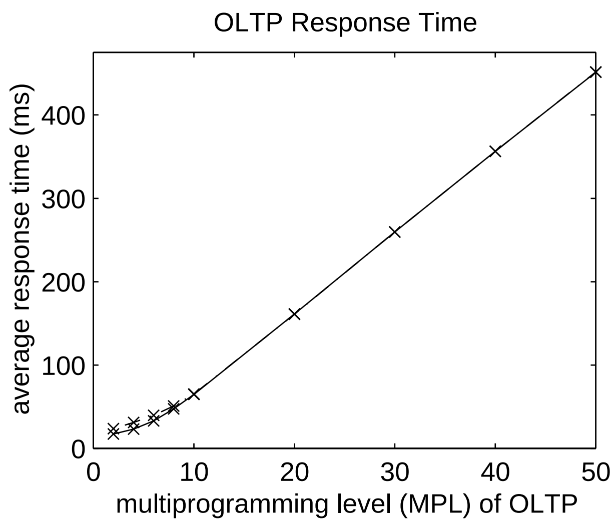
Conclusion & Future Work

On-Disk Scheduling

- **read background blocks only when queue is empty**





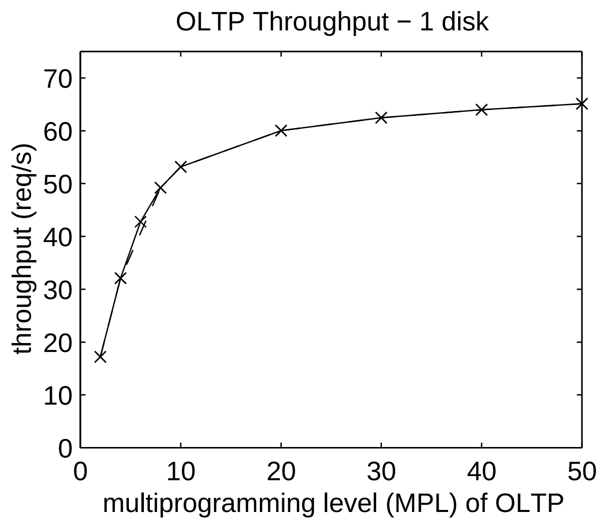


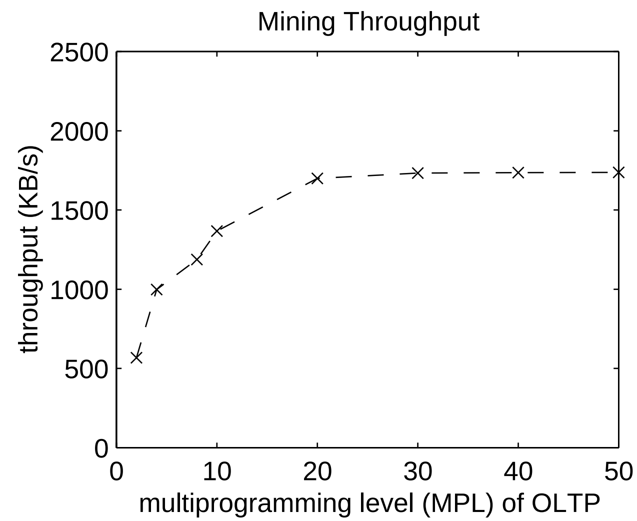
Background scheduling

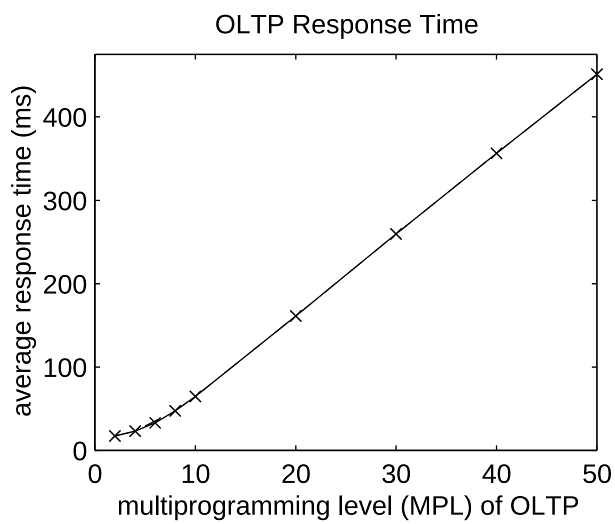
- **vary multiprogramming level - total number of pending requests**
- **background forced out at high foreground load**
- **up to 30% response time impact at low load**

On-Disk Scheduling

- **read background blocks only when completely "free"**





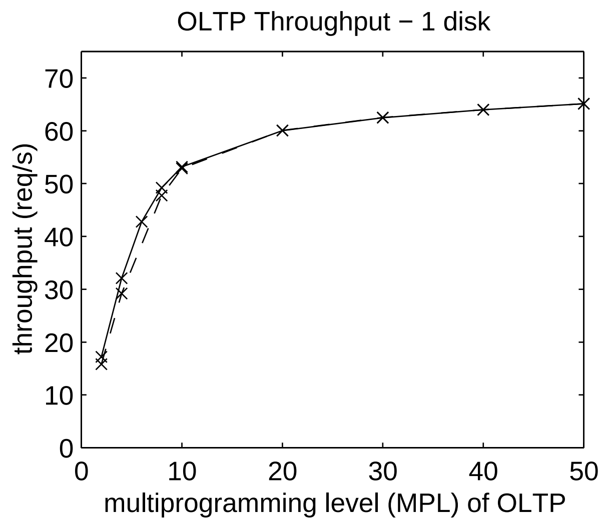


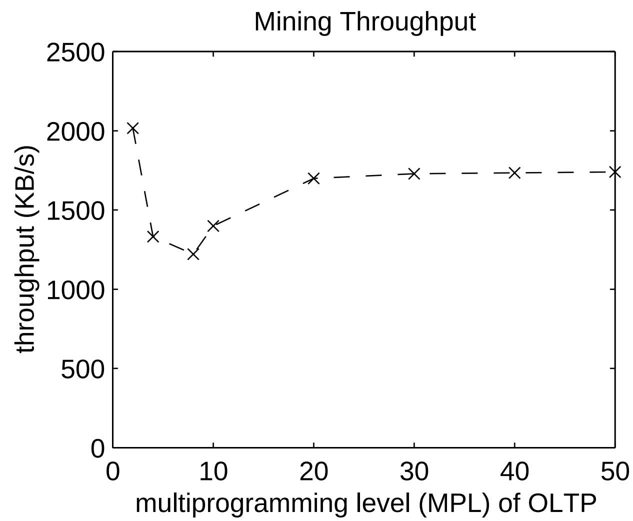
Freeblock scheduling

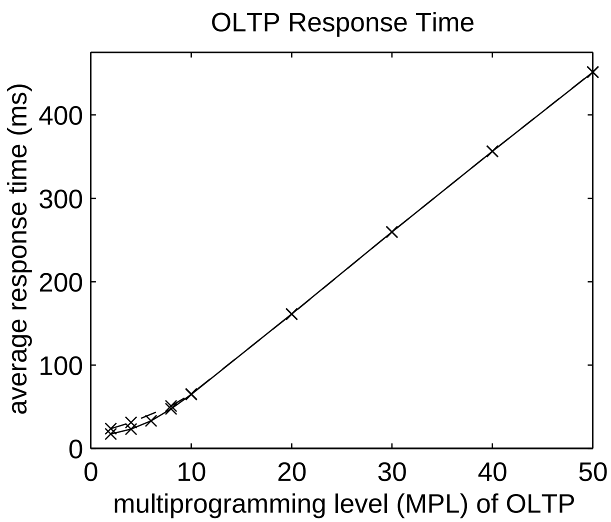
- **opportunistic read**
- **constant background bandwidth, even at highest loads**
- **no impact on foreground respond time**

On-Disk Scheduling

- **combine background and "free" blocks**







Integrated scheduling

- **possible only at drives**
- **combines application-level and disk-level information**
- **achieves 30% of the drive's sequential bandwidth "for free"**

Outline

Motivation

Freeblock Scheduling

Scheduling Trade-Offs

Performance Details

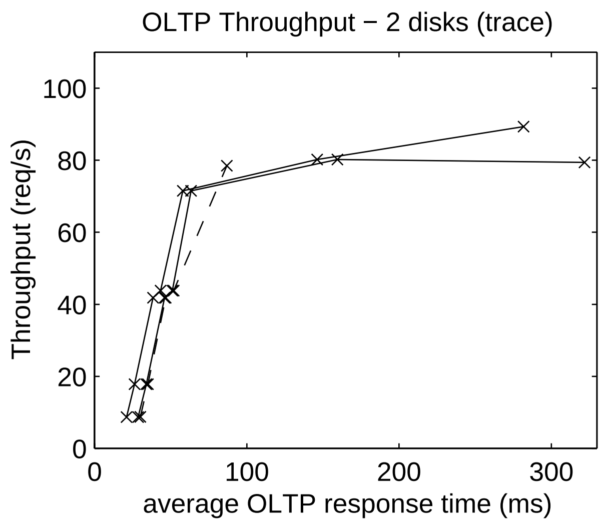
Applications

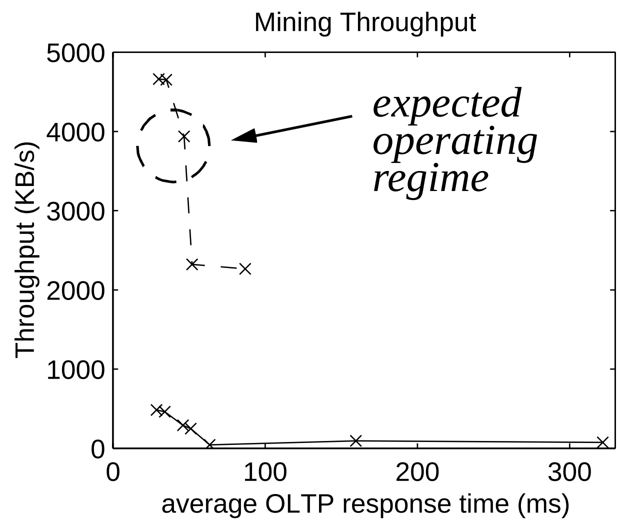
Related Work

Conclusion & Future Work

Validation - Traced Workload

- **using TPC-C disk trace, two disks**



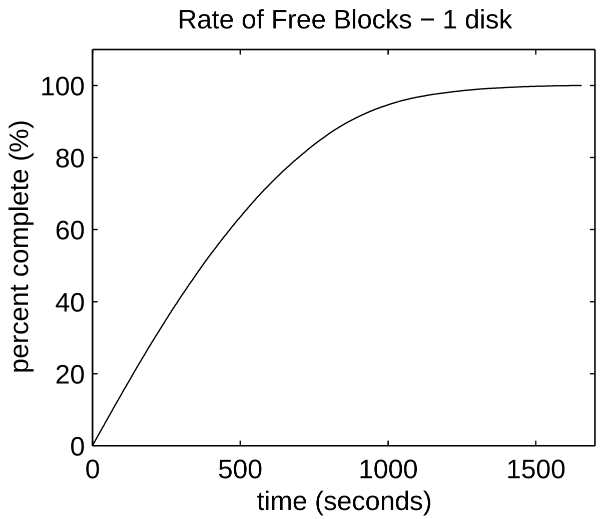


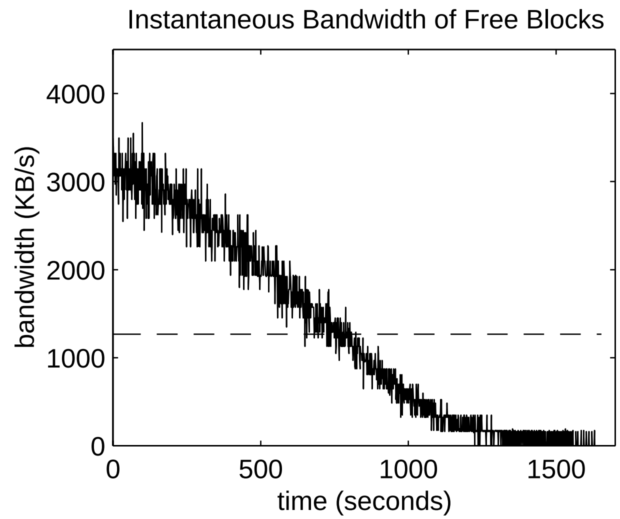
Performance validation

- **predicted benefit possible with real workload**
- **very good performance at “normal” usage**

Freeblock Bandwidth

- *pessimistic* - read the entire disk



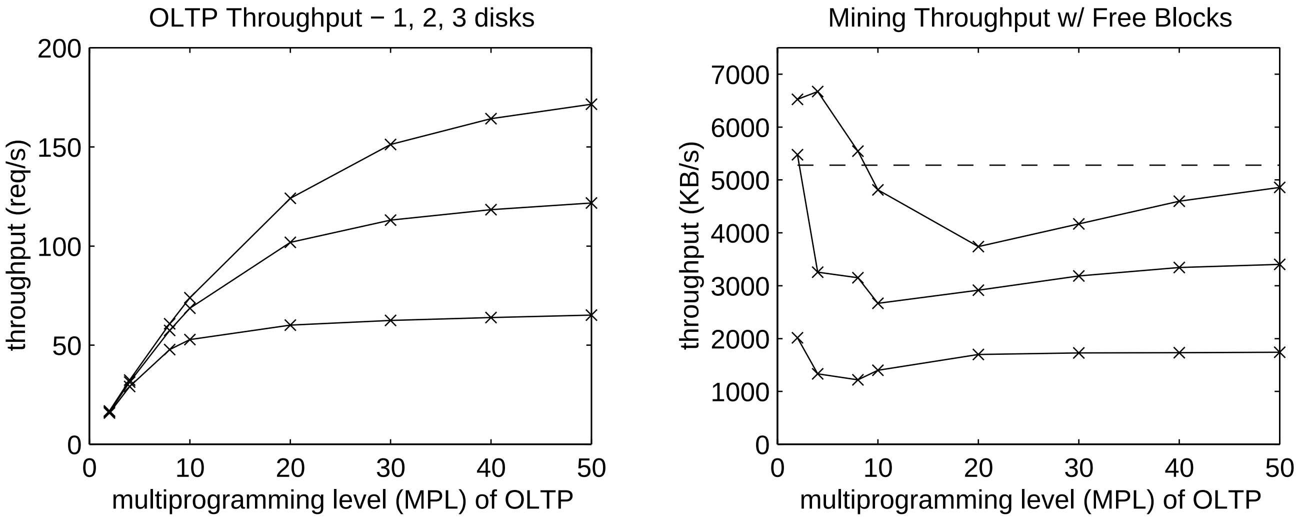


Performance details

- 85% of disk read in 1/2 total time
- bandwidth drops as only “edge” blocks remain
- affected by relative layout of relations on disk

Multiple Disks

- **data striped across multiple disks**



Increase number of disks

- **additive performance, as expected**
- **three freeblock disks equivalent to a single disk "dedicated" to mining**

Outline

Motivation

Freeblock Scheduling

Scheduling Trade-Offs

Performance Details

Applications

Related Work

Conclusion & Future Work

Applications for Freeblocks

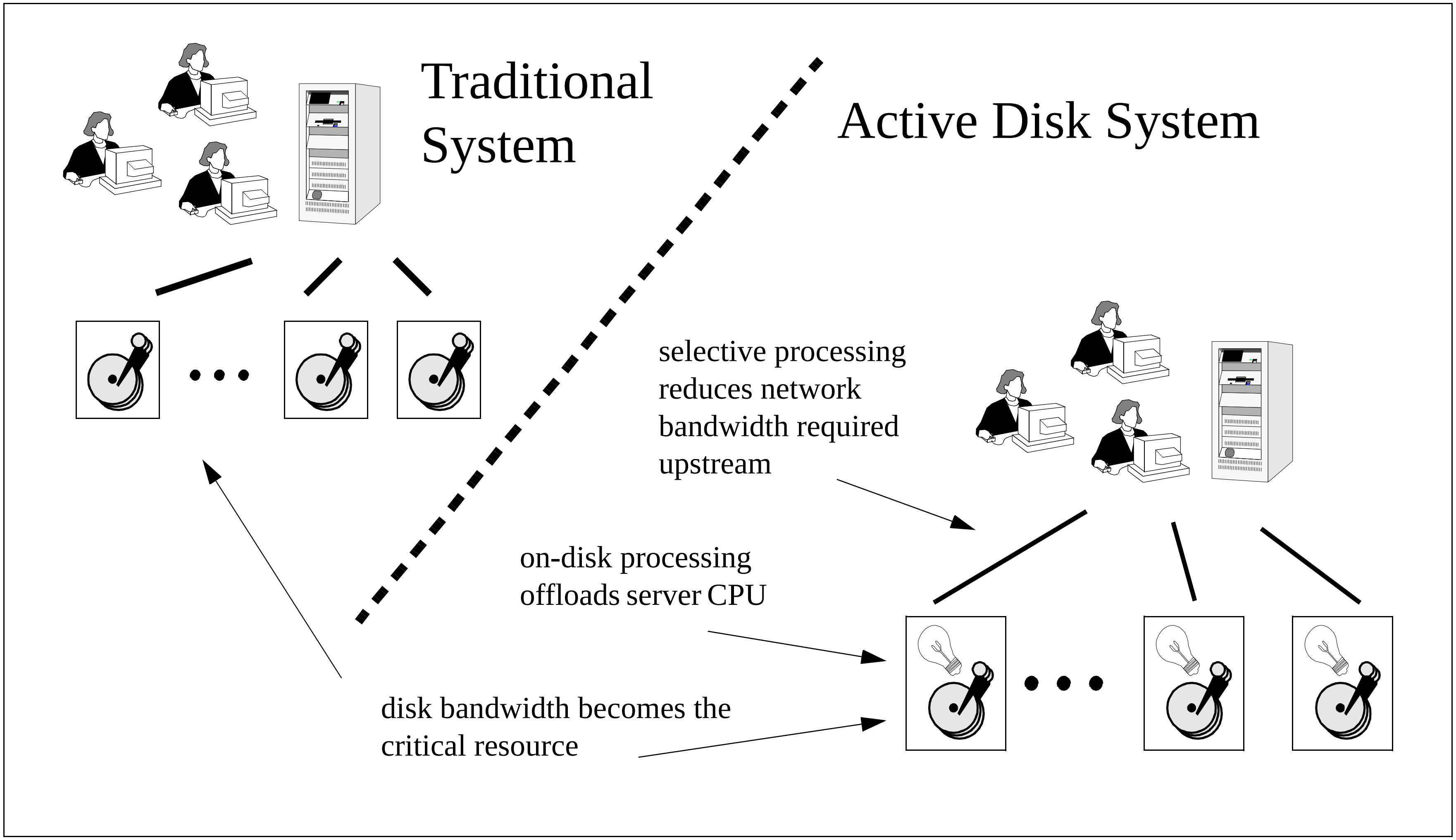
Data Mining for Free

- scans
  - parallel table scans
  - search, association rules, ratio rules & SVD, clustering
- sampling
  - statistical mining, histogram maintenance
  - assuming a slightly modified “random” is acceptable
- assuming that CPU and memory resources are available

Background Utilities

- layout optimization
- incremental backup
- virus scan, fast find
- assures “some” progress, even on busy disks

Synergy with Active Disks



resources required: cpu, memory, network, *disk*

Outline

Motivation

Freeblock Scheduling

Scheduling Trade-Offs

Performance Details

Applications

Related Work

Conclusion & Future Work

On-drive Optimization

Request Scheduling

- **fcfs, scan, look, elevator, sptf**
- ***limited by short queues***

Interface Advances

- **MFM - direct control**
- **SCSI - abstracted interface, fixed size blocks, linear addresses**
  - **cylinder groups, block remapping, ...**
- **current debates - which higher-level interface?**
  - **Network-Attached Storage (NAS)**
  - **Object-Based Disks (OBD)**

How to Get More Information from Applications

- **operating system interfaces limited**
- **“hints” - informed prefetching and caching**
- **Active Disks - push application knowledge to disks**

Related Work

Disk Scheduling

- **studied for many years [Denning67, ..., Worthington94]**

Combined OLTP and Mining

- **memory in mixed workload [Brown92, Brown93]**
  - **multiple workload classes, boundary shifts**
- **OLTP and DSS on same system [Paulin97]**
  - **35% - 100% impact**
  - **disk is critical resource**
- **Sun/IBM benchmark system [TPC97]**
  - **separate CPUs, separate memory**
  - **(mostly) separate disks**

On-disk Optimization

- **zero-latency reads for prefetch**
- **fast writes [Wang99]**

Conclusion & Further Work

Exploit technology trends

- disk bandwidth and positioning time not keeping pace
- use scheduling knowledge at the disks

Novel functionality

- data mining for free - close to 30% bandwidth “for free”
- even at high foreground loads

Interface design

- how to get more information into the disk
- where is the best to place processing resources

Further Work

- details of interface, what file system extensions?
- explore interaction/synergy with data layout
- quantify costs/benefits in a running system

Future Work

Evaluation of All Database Operations

- **optimization for index-based scans**
- **update performance, combine with fast writes**

Programming Model - Application Layers

- **get information through the file system interface**
  - **storage layout**
  - **access patterns**

Implementation Details

- **drive resource requirements**
  - **memory - low**
  - **cpu - medium**
- **demonstrate a “real” background workload**
  - **implement combined OLTP/mining**
  - **or a utility operation**

Extra Slides

Excess Device Cycles Are Here

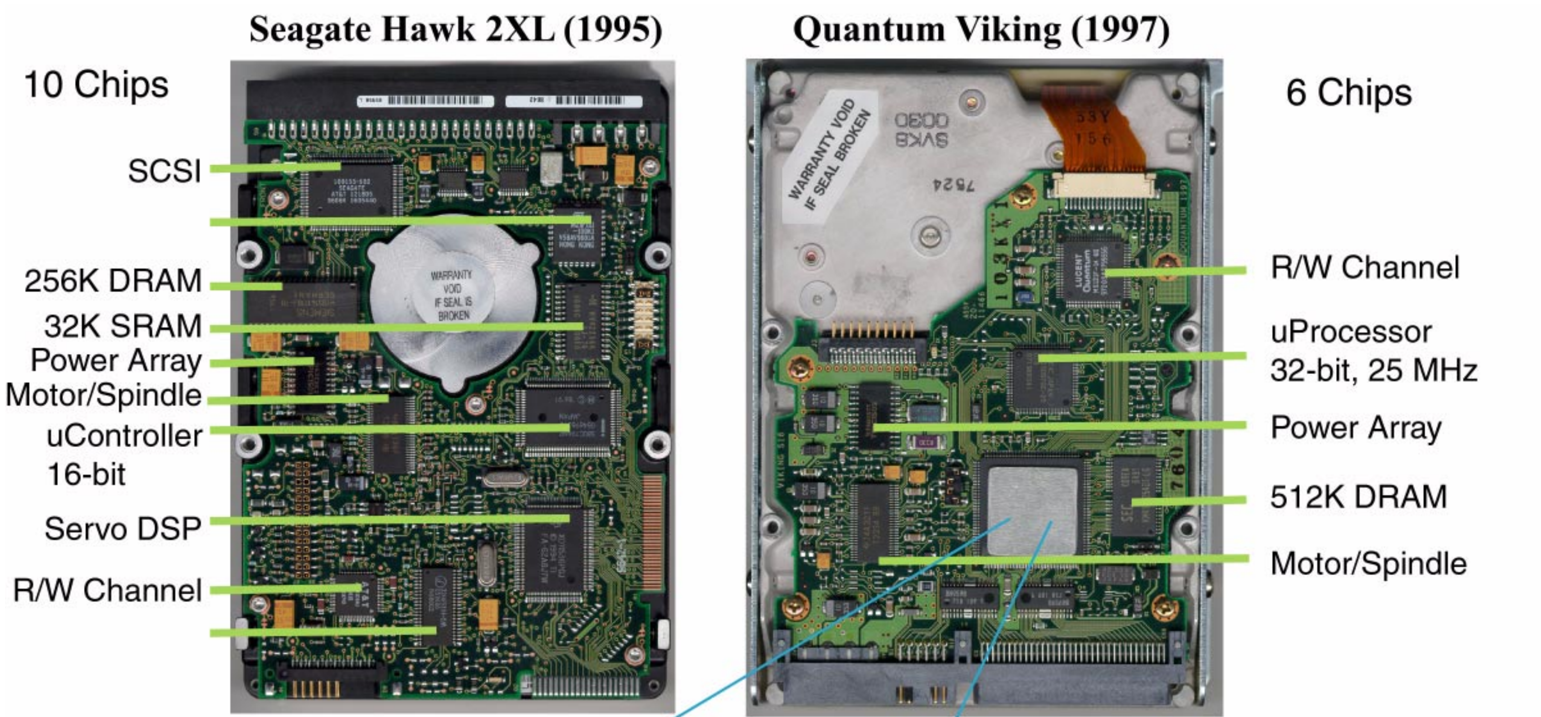
Higher and higher levels of integration in electronics

- specialized drive chips combined into single ASIC
- technology trends push toward integrated control processor
- Siemens TriCore - 100 MHz, 32-bit superscalar today
  - to 500 MIPS within 2 years, up to 2 MB on-chip memory
- Cirrus Logic 3CI - ARM7 core today
  - to ARM9 core at 200 MIPS in next generation

High volume, commodity product

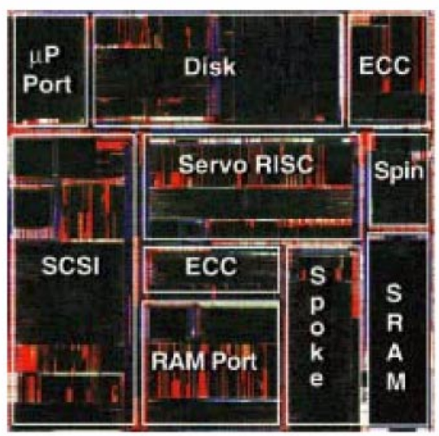
- 145 million disk drives sold in 1998
  - about 725 petabytes of total storage
- manufacturers looking for value-added functionality

Evolution of Disk Drive Electronics

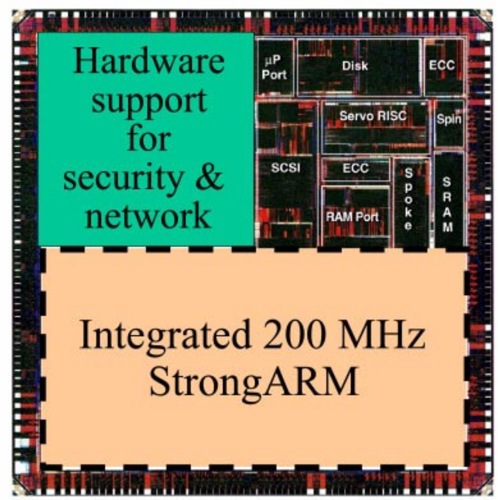


Integration

- reduces chip count
- improves reliability
- reduces cost
- future integration to processor on-chip
- but there must be at least *one* chip



Trident ASIC

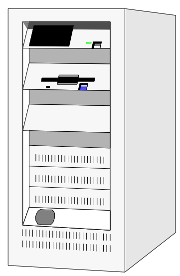


Future Generation ASIC

Opportunity

TPC-D 300 GB Benchmark, Decision Support System

Database Server



Digital AlphaServer 8400

- 12 x 612 MHz 21164
- 8 GB memory
- 3 64-bit PCI busses
- 29 FWD SCSI controllers

= 7,344 total MHz

3 x 266 = 798 MB/s

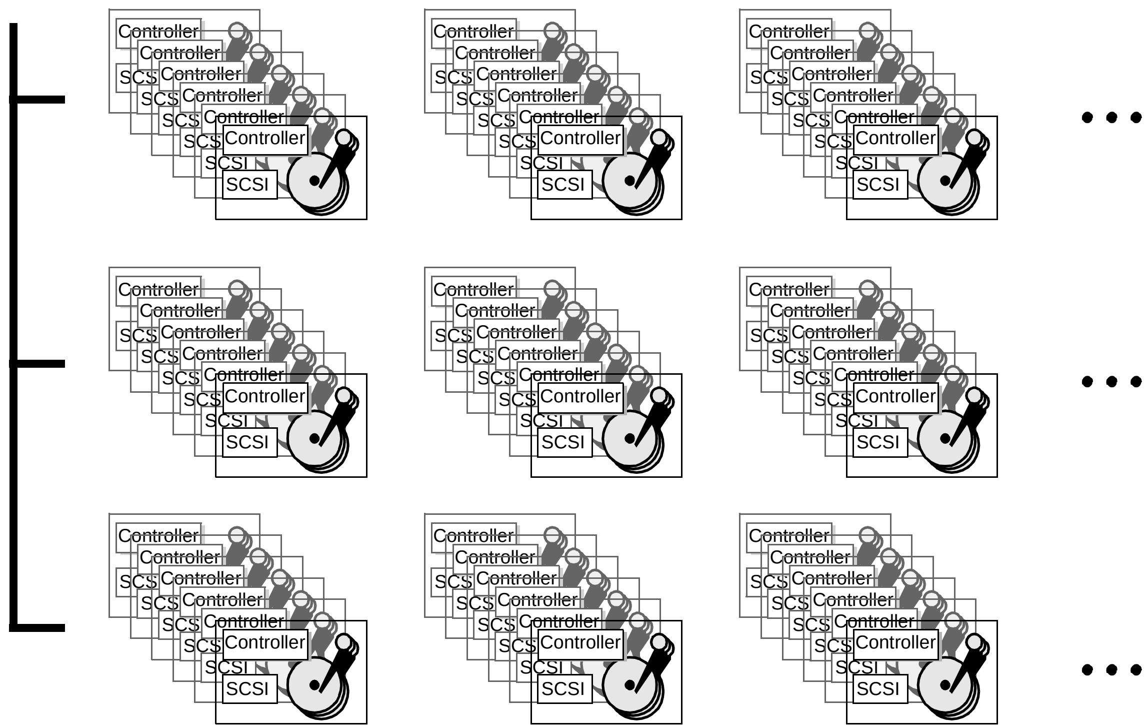
29 x 40 = 1,160 MB/s

Storage

- 520 rz29 disks
- 4.3 GB each
- 2.2 TB total

= 104,000 total MHz
(with 200 MHz drive chips)

= 5,200 total MB/s
(at 10 MB/s per disk)



Advantage - Active Disks

Active Disks **execute application-level code on drives**

Basic advantages of an Active Disk system

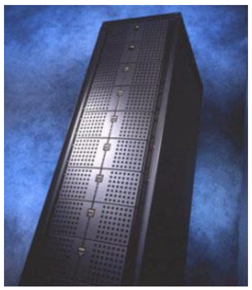
- **parallel processing - lots of disks**
- **bandwidth reduction - filtering operations are common**
- **scheduling - little bit of “strategy” can go a long way**

Characteristics of appropriate applications

- **execution time dominated by data-intensive “core”**
- **allows parallel implementation of “core”**
- **cycles per byte of data processed - *computation***
- **data reduction of processing - *selectivity***

Network "Appliances" Can Win Today





Dell PowerEdge & PowerVault System

Dell PowerVault 650F $46,549 x 12 =558,588
512 MB cache, dual link controllers, additional 630F cabinet, 20 x 9 GB FC disks, software support, installation
Dell PowerEdge 6350 $9,210 x 12 = 110,520
500 MHz PIII, 512 MB RAM, 27 GB disk
3Com SuperStack II 3800 Switch 6,679
10/100 Ethernet, Layer 3, 24-port
Rack Space for all that 20,710

NASRaQ System



Cobalt NASRaQ $1,617 x 240 =388,080
250 MHz RISC, 32 MB RAM, 2 x 10 GB disks
Extra Memory (to 128 MB each) $174 x 240= 41,760
3Com SuperStack II 3800 Switch $6,679 x 11= 76,736
240/24 = 10 + 1 to connect those 10
Dell PowerEdge 6350 Front-End 9,210
Rack Space (estimate 4x as much as the Dells) 82,840
Installation & Misc 50,000

Comparison

	Dell	**Cobalt**
Storage	2.1 TB	4.7 TB
Spindles	240	480
Compute	6 GHz	60 GHz
Memory	12.0 GB	30.5 GB
Power	23,122 W	12,098 W
Cost	$696,794	$648,626