Characterizing I/O-intensive Workload Sequentiality on Modern Disk Arrays

Kimberly Keeton, Guillermo Alvarez,
Erik Riedel, and Mustafa Uysal
Hewlett-Packard Labs
Storage Systems Program
{kkeeton,galvarez,riedel,uysal}@hpl.hp.com

4th Workshop on Computer Architecture Evaluation using Commercial Workloads
January 21, 2001

Motivation

- **SSP goal: develop analytic models of storage devices to predict workload performance**
  - **Finding: even moderately sophisticated models give insufficient accuracy. Why?**
- **Why?**
  - **New features for mid-range and high-end disk arrays have significant impact on performance**
  - **Real-world workloads exhibit complex behavior**
- **Issues:**
  - **We don't sufficiently understand how array features perform for complex workloads**
  - **Our workload characterizations don't have attributes to capture effectiveness of features**

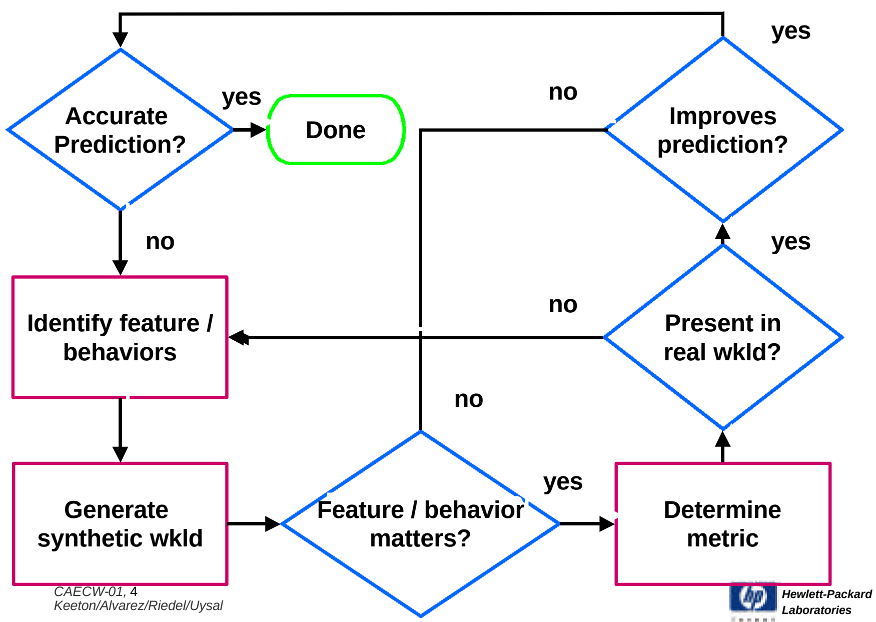
Key high-level question

- **Goal: accurate performance prediction**
- **Need:**
  - **Model of important disk array features**
  - **Model of important workload behaviors**
- **Approaches:**
  - **Identify and quantify new array features/workload behaviors (this talk)**
  - **Quantify relevance of workload metrics by using them to synthetically generate workloads (Kurmas, et al.)**

Outline

- **Motivation**
- **Description of our approach**
- **Experimental infrastructure**
- **Prefetching results**
  - **Synthetic workloads**
  - **Real workloads**
- **Conclusions and ongoing/future work**

Approach: find and quantify features/behaviors



Experimental infrastructure: HP FC-60 disk array

Characteristic	*Our configuration*	*Max configuration*
Cache size	256 MB x 2	512 MB x 2
Disk enclosures	6	6
Disks per enclosure	5	10
Capacity per disk	18 GB	73 GB
Total capacity	0.5 TB (unprotected)	4.4 TB (unprotected)

- Two controllers
  - One fibre channel port/controller
  - Each connected to all enclosures
- Six ultra-wide SCSI interfaces to controllers (40 MB/s each)
- Cache: split between controllers
  - Write-back policy, with writes mirrored across controllers

FC-60 performance

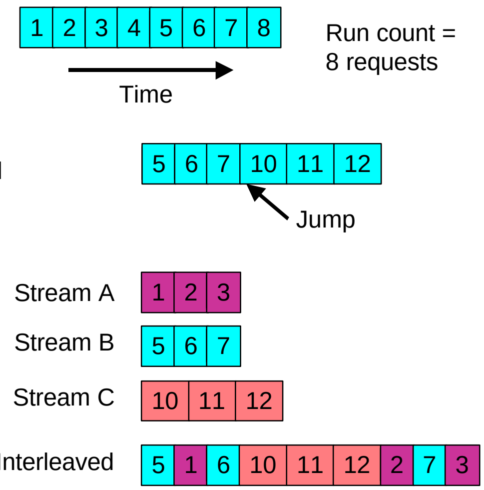
- Environment:
  - Single 4-disk R5 LUN
  - 16KB stripe unit size
  - Synthetic workload generation
- Random performance:
  - 2KB random reads : 381.76 IOPs
  - 256KB random reads : 17.6 MB/s
- Sequential performance:
  - 2KB sequential reads: 3656.4 IOPs
  - 256KB sequential reads : 40.3 MB/s
- Top observed performance per controller: 84 MB/s, 11,000 IO/s

Prefetching features

- Both arrays and disks do prefetching
- Disk prefetching
  - Track buffer’s worth
  - Up to disk controller cache segment
- Array prefetching
  - Along a logical unit (LUN) – stripe group
  - Per-request minimum, maximum prefetch
  - Multiple of request size
- Effective for strictly sequential workloads
- Effective for other workload behaviors?

Workload sequentiality behavior

- **Strict sequentiality**
  – Simple run count insufficient
- **Workloads exhibit pseudo-sequential behavior**
  – "Holes" or jumps in sequential runs
  – Interference between streams



CAECW-01, 8
Keeton/Alvarez/Riedel/Uysal

Hewlett-Packard Laboratories

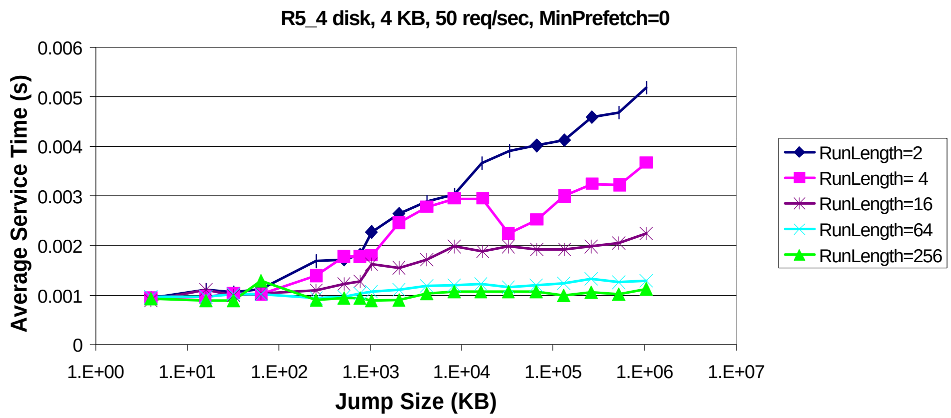
Synthetic experiments to focus on behaviors

- **How well does prefetching work for pseudo-sequential behavior?**
- **Evaluation criteria**
  – Performance = average array service time
    - includes no device driver queueing
- **First set: jumps in sequential runs**
- **Second set: interleaved streams**
- **Experimental parameters**
  – 4-disk R5 LUN, 16KB stripe unit size
  – 4 KB requests, 50 requests/sec
  – Short run counts (4 requests) vs. long run counts (256 requests)

CAECW-01, 9
Keeton/Alvarez/Riedel/Uysal

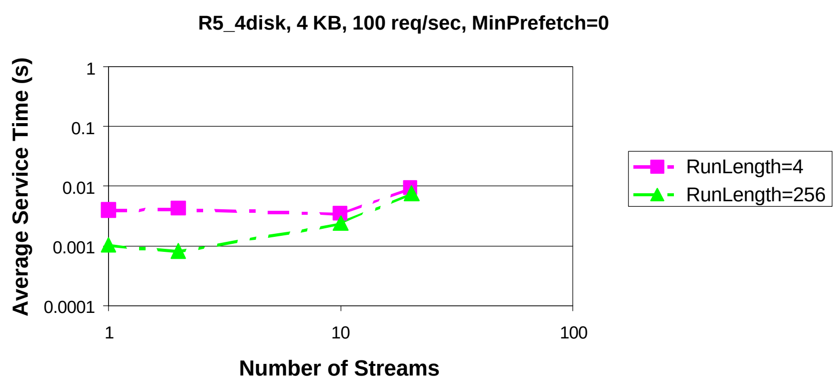
Hewlett-Packard Laboratories

Impact of holes on service time?



- Large runs have consistently low service time
- Short runs with small jumps (< 64KB) have low service time

Impact of interfering streams on service time?



- Constant aggregate request rate = 100 requests/sec
- After threshold of ~10 streams, service time increases as # streams increases

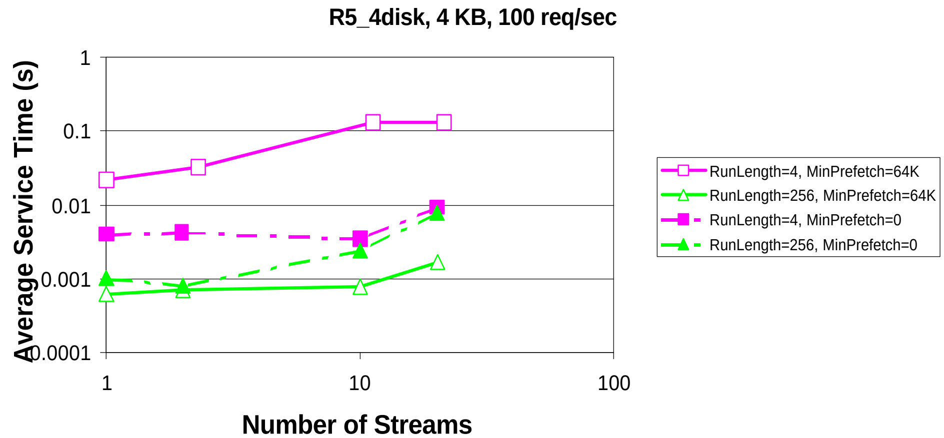
Predicting prefetching effectiveness

- **Expectations:**
  - Prefetching effective for short jumps
  - Prefetching effective for small number of interfering streams
- **Evaluation methodology:**
  - Ideally, compare array with prefetching disabled vs. prefetching enabled – control knobs unavailable
  - Compare workload performance with no forced array prefetching vs. forced minimum array prefetching
    - *MinPrefetch* = 0, 64K
  - Performance = average array service time
    - includes no device driver queueing

Prefetching success for jumps?

- **Prefetching successful for:**
  - small jumps (< ~1MB), for both short and long runs

Prefetching success for interfering streams?



- Prefetching effective for long runs
- Prefetching detrimental for short runs

How to measure workload behaviors?

- Proposed metrics for quantifying sequentiality behavior:
  - Jumps/holes in sequential runs
    - *Jump distance* = "gap" between next address and expected next address if accesses are sequential

5	6	7	10	11	12

Jump distance = 2

  - Interleaved accesses from different streams
    - *Interference frequency* = number of requests from other streams that occur between successive requests from this (sequential) stream

Interference frequency: 1 4

5	1	6	10	11	12	2	7	3

4 1

Real applications

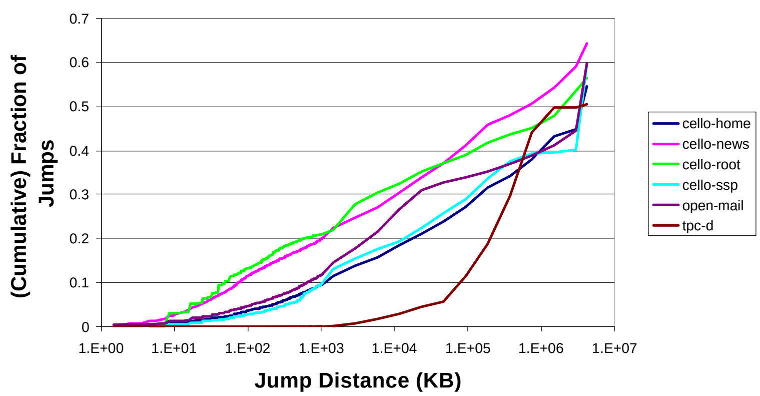
- **Questions**
  - How big are the jumps in these workloads?
  - How much stream interference occurs in workloads?
  - Is there a single unified sequentiality metric?
- **Replay traces from full-scale real workloads:**
  - 300-GB TPC-D: table, index, temp, log, summary
  - Open Mail email server
  - Cello file server traces: root, news, home, ssp
- **Start with single-LUN experiments**
  - Pick a representative array, and a representative LUN

Do workloads have jumps?

Workload	*P(backward jump)*	*P(sequential)*	*P(forward jump)*
cello-home	0.40	0.11	0.49
cello-news	0.35	0.03	0.62
cello-root	0.19	0.56	0.25
cello-ssp	0.06	0.85	0.09
open-mail	0.38	0.06	0.56
tpc-d	0.48	0.02	0.49

- **Forward jumps comprise ~half of accesses for most workloads**
- **Cello-root and cello-ssp are mostly sequential**

How big are workload spatial jumps?



- Most workloads have a range of small and large jumps
- tpc-d has only large jumps

Do workloads have interleaved streams?

Workload	***Sequential Run Count (old)***	***Interference Frequency***	***Detangled Run Count***	***Relaxed Detangled Run Count***
cello-home	1.12	124.6	1.49	3.29
cello-news	1.03	98.5	1.12	20.40
cello-root	2.23	29.0	2.71	9.40
cello-ssp	6.74	28.9	20.46	40.86
open-mail	1.06	7.4	1.32	3.98
tpc-d	1.02	14.3	21.07	21.30

- Multi-threaded workloads (tpc-d) have interfering sequential streams
- Difficult to discern interleaved sequential streams for workloads with more random behavior (cello-{home, news, root})

Summary of results

- Simple run count spatial locality metrics too restrictive to capture prefetch-friendly behavior
- New workload characteristics to be captured:
  - "Holes" or jumps within a stream: jump distance
  - Interleaved accesses from other streams: interference frequency
- Array prefetching effective for:
  - Longer sequential runs
  - For shorter runs, smaller jumps between runs
  - Small number of streams (low interference frequency)
- Working on proposal for new run count metric to incorporate these characteristics

Ongoing/future work

- Metric incorporating all factors doesn't predict as well as expected
- Other confounding features/behaviors to consider:
  - Burstiness
  - Cache capacity and interference
  - Write destaging
  - Stream detection
  - Degraded mode
- Ultimately:
  - Incorporate prefetching and other findings into our analytic array models and workload specifications

Conclusions

- Developed methodology for evaluating how array features and workload behaviors impact performance
  - Identify several potential prefetching features and corresponding workload behaviors
  - Use synthetic workloads to isolate effects
  - Propose metrics for measuring prefetching behavior in real workloads
  - Measure proposed metrics on traces replayed from real workloads
- Simple sequentiality metrics provide insufficient predictive power
- Additional behaviors to consider
  - Jump distance and interference frequency

Backup slides

Related work

- Analytical modeling approaches
  - Modeling prefetching in disks [Shriver…98]
- Studies of array performance characteristics
  - Few on the real thing: [Chen…90],[ChervenakKatz91]
  - Conseqs of data distribution: layouts [LeeKatz93],[Alvarez…98]
- How arrays can take advantage of application characteristics
  - Autoraid [Wilkes…96]
  - Setting storage system params:[ChenLee95],[ChenPatterson90],[ShenoyVin97],[Jacob96]
  - Commercial products: [EMC's SRDF], [XP's moral eqv]
- How to present requests to devices, given a workload
  - prefetching
    - informed [Tomkins…97], [Kimbrel…96]
    - uninformed [Chervenak…99]