I/O Characterization of Commercial Workloads

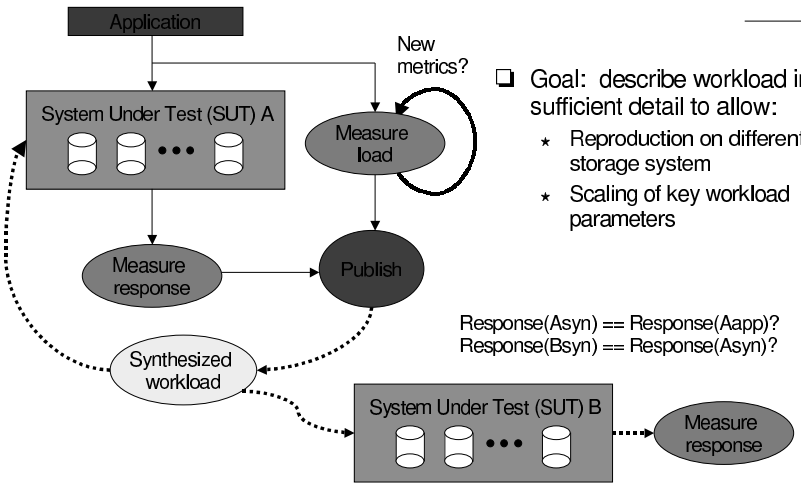
Kimberly Keeton, Alistair Veitch, Doug Obal,
and John Wilkes
Storage Systems Program
Hewlett-Packard Laboratories
www.hpl.hp.com/research/itc/csl/ssp
kkeeton@hpl.hp.com

Third Workshop on Computer Architecture Evaluation
using Commercial Workloads (CAECW '00)
January 9, 2000

Why study I/O characteristics?

- It's interesting:
  - Many commercial applications are I/O-intensive
  - Commercial apps have different characteristics than the ones we already understand [SOSP99]
- It's necessary:
  - Online monitoring of system operation/performance
  - Offline monitoring to determine opportunities for improvement
  - Input to storage system design process
    - "What if" design questions
    - Predicting effects of new or "scaled" workloads
  - Generation of representative synthetic workloads
    - Test performance of new designs
    - Compare systems

How to study I/O characteristics?



- Goal: describe workload in sufficient detail to allow:
  - Reproduction on different storage system
  - Scaling of key workload parameters

Response(Asyn) == Response(Aapp)?
Response(Bsyn) == Response(Asyn)?

Outline

- Motivation
- Understanding application I/O characteristics
- Case studies:
  - Electronic mail server
  - TPC-D-based decision support database server
- Ongoing investigations
- Conclusions

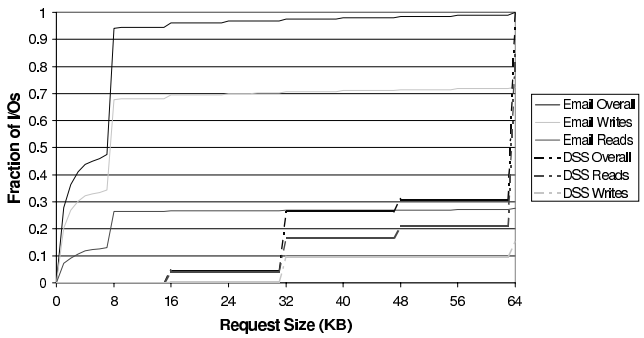
Understanding Application I/O Characteristics

- Lessons learned:
  - List of important characteristics is longer than you think
  - Distributions, not averages, are important
- Characteristics of interest:
  - Request size distribution
  - Request rate distribution
  - Read:write ratio
  - Spatial locality (e.g., sequentiality)
  - Temporal locality (e.g., data re-references)
  - Correlation between accesses to different parts of storage system
  - Burstiness
  - Phased behavior

Case Studies

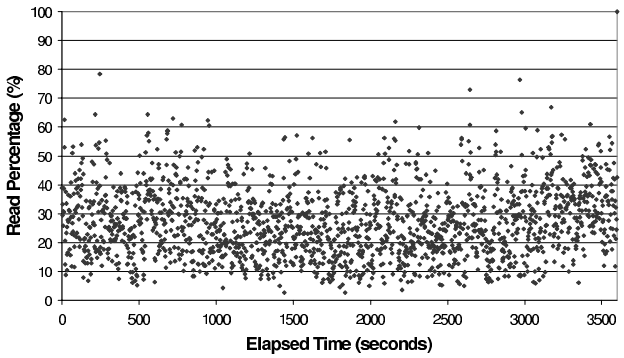
- Electronic mail server
  - HP OpenMail
  - Peak operation period
  - ~1400 active users
- Decision support database server
  - Oracle
  - 300 GB TPC-D database
  - Presentation focus: TPC-D Q5

Request Size Distributions



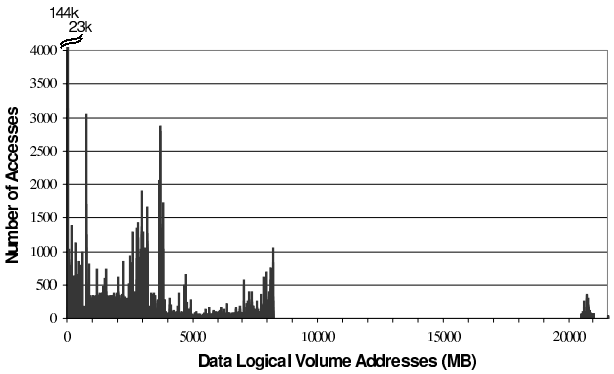
- Email dominated by small (<= 8 KB) writes
- DSS dominated by larger (64 KB) reads

Email: Read Percentage



- Average read percent: 28%

Email: Access Locality

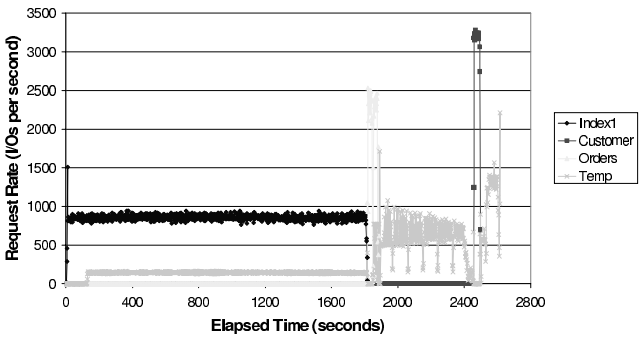


- Beginning of address range heavily accessed

Email Observations

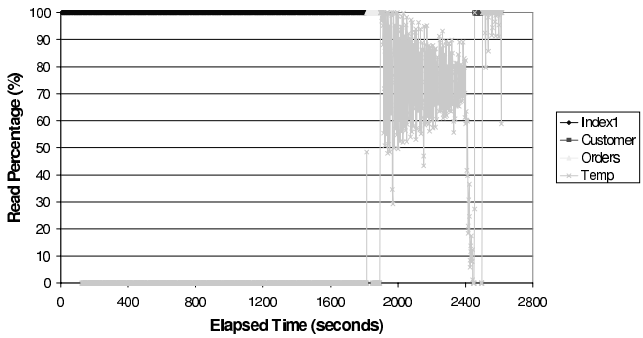
- Small (<= 8 KB) writes dominate
  - RAID 5-based storage suboptimal
- Highly localized write operations
  - Disk array caching important for higher performance
- Traces from this and other OpenMail servers will be contributed to Storage Performance Council effort to develop standard I/O benchmark based on electronic mail.

I/O Phasing: DSS Database Request Rates



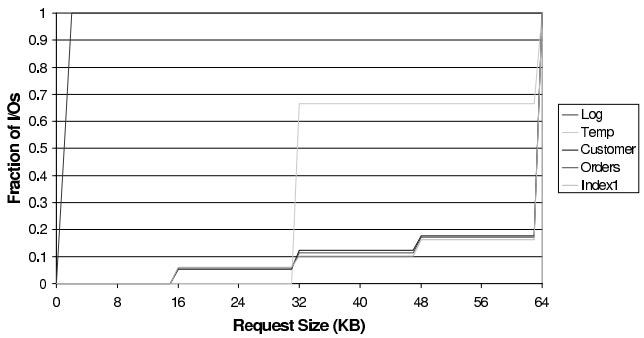
- Most multi-table queries have multiple phases

I/O Phasing: DSS Database Read Percentage



- "Read-only" workload exhibits some writes

DSS DB Request Size Distribution



- Different behavior from tables/indices vs. temp vs. log

DSS DB Observations

- DSS queries exhibit phased I/O behavior
  - Different tables active at different periods in the query
  - Storage system design depends on correlation between these streams
  - How will multiple simultaneous query streams impact DSS storage behavior?
- "Read-only" decision support isn't
  - Temporary table writes not uncommon in complex queries
- Different application storage units (e.g., table vs. index vs. log) have different access patterns
  - Device design (e.g., stripe unit size) should be optimized accordingly

Ongoing Investigations

- How best to express:
  - Spatial locality: run count vs. jump distance vs. ?
  - Temporal locality: unique re-reference distance
  - Correlated activity: stream interleaving
  - Short-term burstiness
- Rules of thumb for scaling these commercial applications?
  - Data set sizes, number of users, query complexity
  - Possible to scale back I/O requirements and observe representative behavior?
- What are the (other) interesting applications?

Conclusions

- Workload I/O characterization drives storage system design at multiple levels
- Need to explore further than simple metrics
  - Iterate to refine the list
- Need distributions, not just averages
- Push the characterization cycle further
  - Generate and measure synthetic workloads based on characterization
- Lots of interesting questions to be answered!

Want to help?

- How can you help?
  - Collect additional traces of new I/O-intensive apps
  - Analyze workload characteristics and publish results
  - Develop new analysis techniques and/or characterization metrics
- SSP tools available to qualified researchers:
  - Rubicon – workload analysis tool
  - Pylon – synthetic workload generator
  - Rome – extensible language for specifying workloads
  - Some traces (Unix file system, etc.)
- For more information:
  - `www.hpl.hp.com/research/itc/csl/ssp/`