Discovery of Application Workloads from Network File Traces

Neeraja J. Yadwadkar, Chiranjib Bhattacharyya, K. Gopinath, Thirumale Niranjan and Sai Susarla

Presented by Daniel Margo and Onur Yoruk

The Semantics of I/O Operations

- Identifying applications by file system I/O.
- E.g. “This I/O pattern corresponds to a `make`”
- Useful for:
  - Tweaking online performance (adaptive systems)
  - Digital forensics / systems administration
  - Helpful for design, debugging, benchmarking
- Lots of research on data mining I/O traces...
- ...but ID-ing applications by I/O pattern is hard.

Why is ID-ing Apps Hard?

- Because I/O traces look like this! →
- I/O is variable, even for a given application.
- Don't know when application “starts”.
- Apps are multiplexed; their I/O is interwoven.

cp * dir/

```
GETATTR Call, FH:0x0eb18814
READDIRPLUS Call, FH:0x0eb18814
READDIRPLUS Reply (Call In 9) ...
LOOKUP Call, DH:0xe003db8b/tqslwiz.h
LOOKUP Reply Error:NFS3ERR_NOENT
GETATTR Call, FH:0x21b1a714
ACCESS Call, FH:0x21b1a714
CREATE Call, DH:0xe003db8b/tqslwiz.h
SETATTR Call, FH:0x6bd9e67c
GETACL Call
GETATTR Call, FH:0x6bd9e67c
READ Call, FH:0x21b1a714 ...
WRITE Call, FH:0x6bd9e67c ...
COMMIT Call, FH:0x6bd9e67c
GETATTR Call, FH:0xe003db8b
LOOKUP Call, DH:0xe003db8b/TrustedQSL.spec
LOOKUP Reply Error:NFS3ERR_NOENT
GETATTR Call, FH:0x2fb1a914
ACCESS Call, FH:0x2fb1a914
CREATE Call, DH:0xe003db8b/TrustedQSL.spec
SETATTR Call, FH:0x65d9e87c
GETATTR Call, FH:0x65d9e87c
READ Call, FH:0x2fb1a914 ...
WRITE Call, FH:0x65d9e87c ...
COMMIT Call, FH:0x65d9e87c
LOOKUP Call,
DH:0xe003db8b/TrustedQSL.spec.in
LOOKUP Reply Error:NFS3ERR_NOENT
```

Problem Statement

- For each application, build an I/O model.
  - Must be resilient to variability.
- Given an I/O trace, search for model match.
  - Must be tractable! (Much easier if not).
  - Must be resilient to interweaving.
- A hard model-to-subset matching problem.

Problem Re-Statement

- Suppose we want to find if `cp`:

`LOOKUP LOOKUP GETATTR ACCESS...`

- Is present in a trace like:

`READ WRITE COMMIT GETATTR`

`LOOKUP LOOKUP CREATE GETATTR`

`ACCESS SETATTR GETATTR READ...`

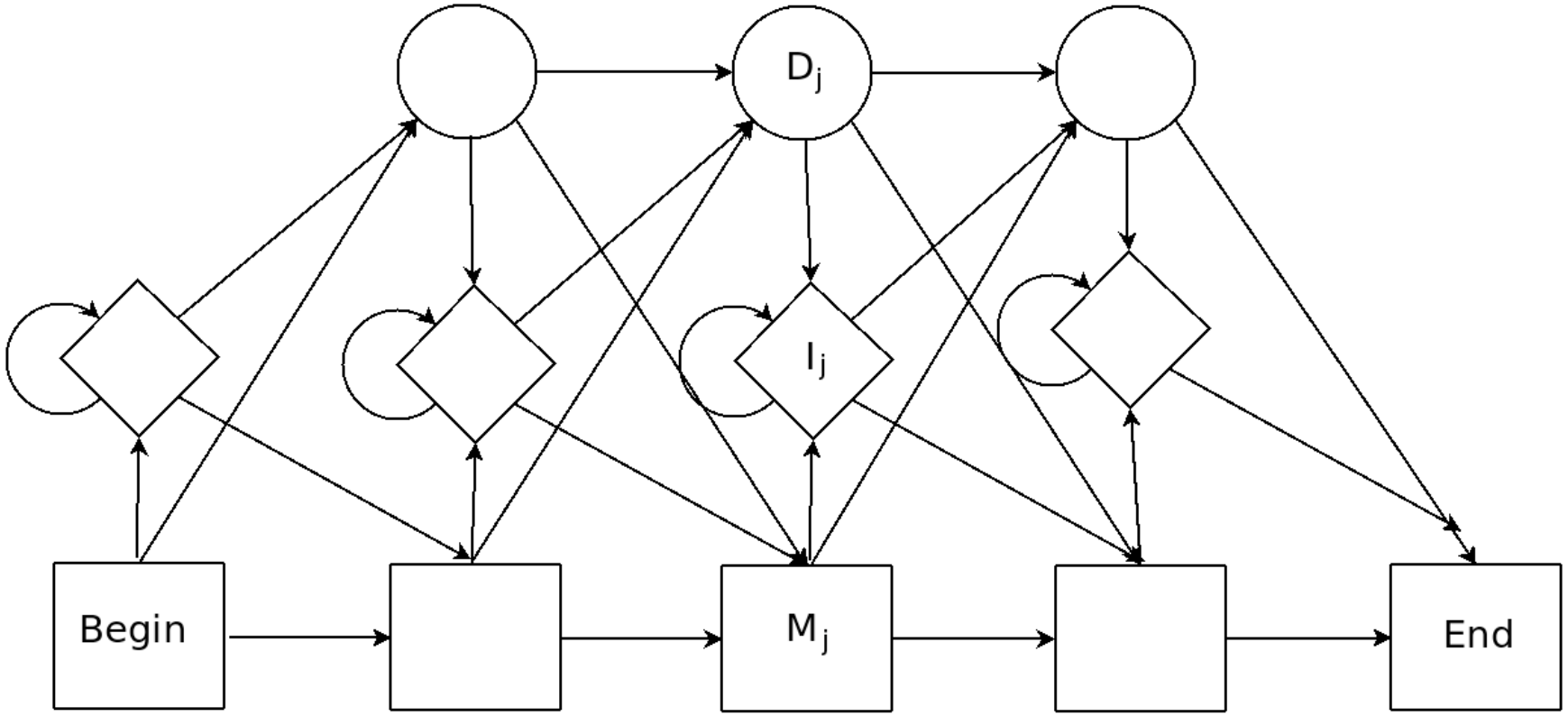
- Let's abbreviate: `LLGA` in `RWCGLLCGASGR`
- This is starting to look a lot like...

DNA Pattern Matching

- Biologists face similar problems with DNA.
  - Need to find known patterns in large strands.
  - Patterns vary (damage, mutation).
  - Patterns are not clearly demarcated.
  - No exact analog of interweaving, unfortunately.
- Many off-the-shelf tools exist for this purpose.
  - Profile hidden Markov models (PHMM).

Profile HMM

This is the kind of thing that can only be shown by example, so let's break out the white board.



Evaluation

- Evaluate on popular UNIX commands:

```
tar untar make edit copy move grep find compile
```

- ~15,000 files / ~1500 directories / up to 7 layers
- And TPC-C workloads (standard FS workload mimicking the activities of a wholesale supplier)
- NFS Packet Traces”
  - Opcodes obtained by filtering out the data portion of the NFS operations.
  - Captured using the Wireshark tool.

Evaluation (2)

- Run each individual workload many times.
  - With different parameters
- Capture their traces
- Train profile HMMs 'til it stops improving (~10 traces).
- Test on remaining traces (cross-validation).
- More precise methodology in paper.

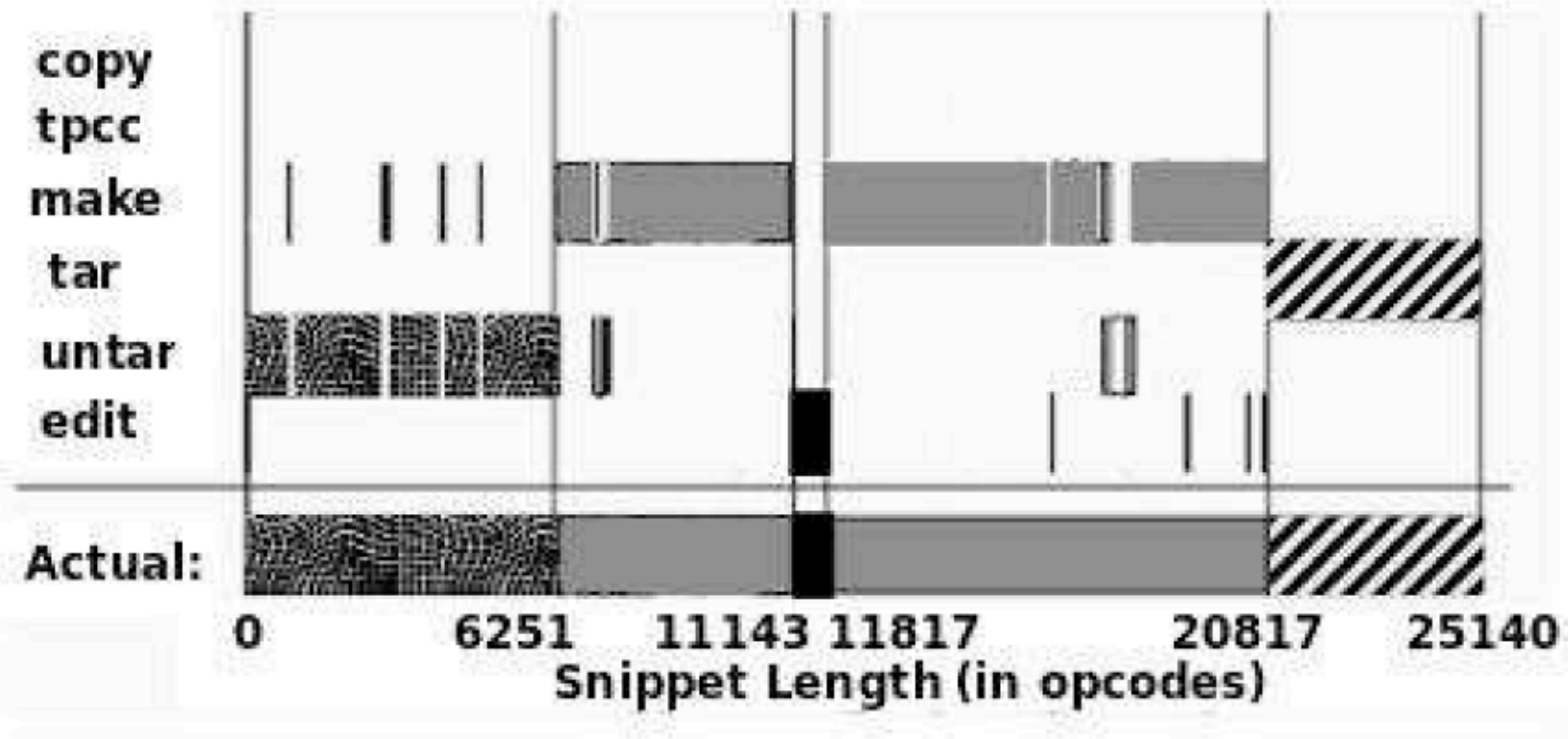
Results: Distinguishing Traces

Confusion matrix:

Trace Command	Models								
	make	**find**	**grep**	**tar**	**untar**	**copy**	**move**	**edit**	**tpcc**
make	**91.7**	1.2		1.2	2.4	3.6			
find		**91.8**	2.1			3.1	1		2.1
grep	1		**72**	22	5				
tar				**100**					
untar				1.2	**98.8**				
copy		1	1		6	**82**	1	9	
move		5.6	0.8	0.8		2.4	**89.6**	0.8	
edit								**100**	
tpcc									**100**

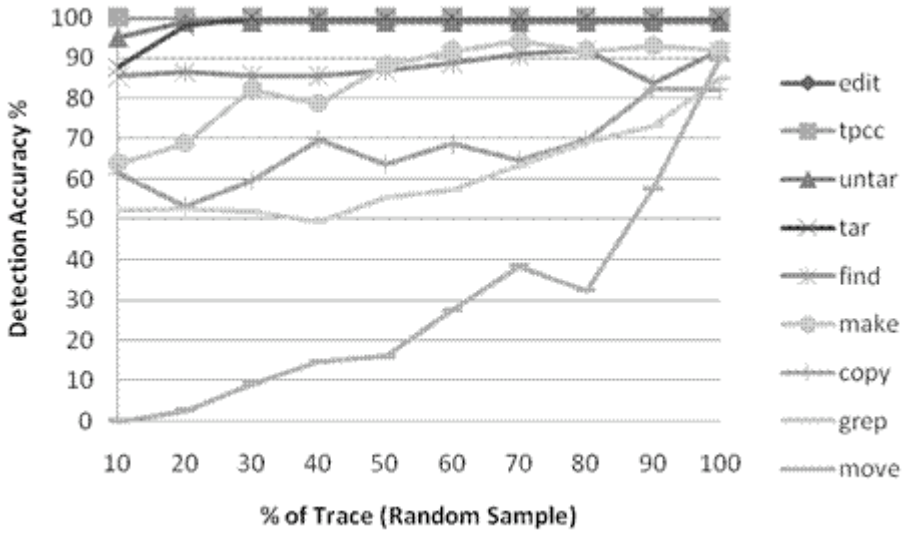
Results: Trace Annotation

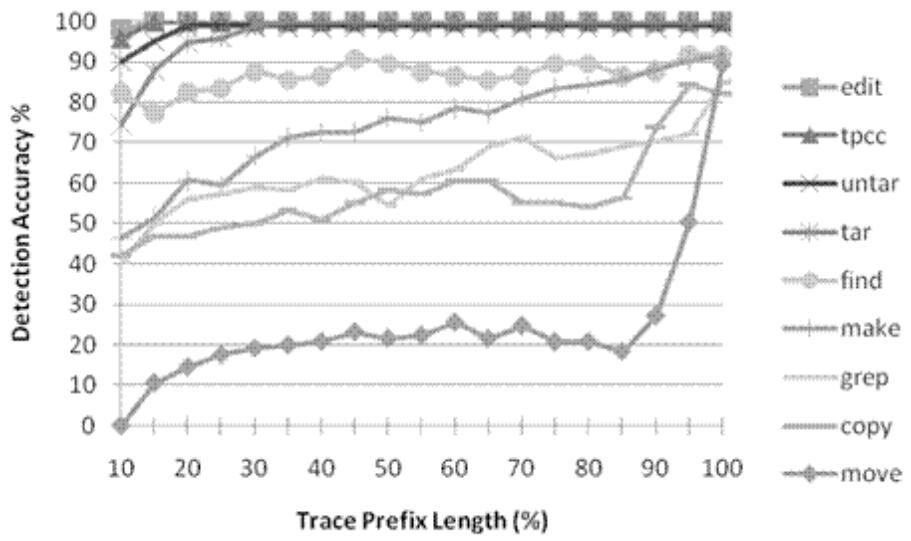
- Annotation Diagram:



Results: Partial Traces

- Left: Accuracy vs. % of partial trace (Random)
- Right: Accuracy vs. % of partial trace (Prefix)

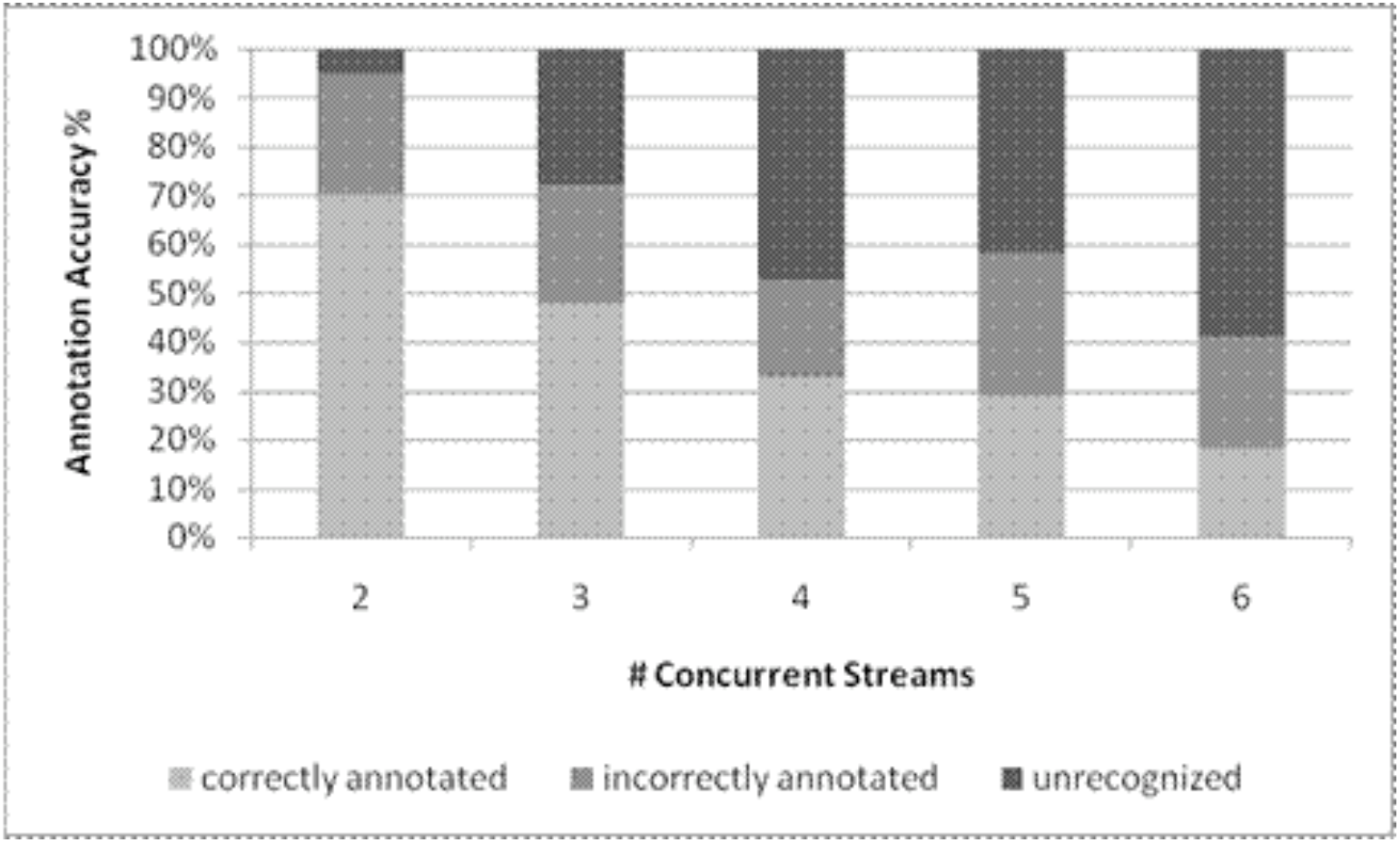




(Never make graphs like these, ever).

Results: Concurrent Traces

- Accuracy tanks quickly beyond ~2 workloads.



Limitations & Conclusion

- Training requires diverse and representative sample of workloads
- Profile HMM is too slow for online analysis.
- NFS trace contains other useful information
  - File names, handles, file offsets, read/write lengths, error responses.
- Same methods can be applied to
  - Network messages, system call traces, disk traces, function call graphs, etc.