

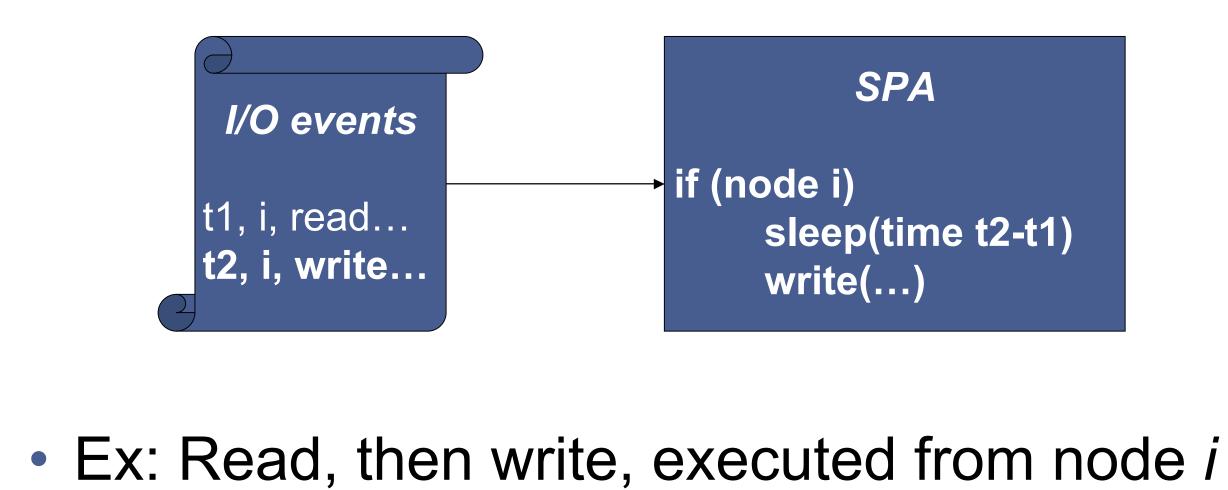
# Measuring I/O Performance in Scalable File Systems Rosie Wacha Darrell Long

### Motivation

- I/O workloads are essential to storage research
  - Measure utility of new algorithms and designs
  - Highlight real performance problem areas
  - Benchmarks can focus on artificial workload types
- Real applications can't be released
  - Proprietary algorithms
  - Require certain hardware
- Can't run in many programming environments
- Want to understand application workloads more generally
  - How would an application run on a different system?
  - Can we classify applications into classes based on I/O workload?
  - Distinguish between system and workload characteristics

## Synthetic Parallel Application

- Mimic the user I/O of a single run of a real parallel application in a C++ program • Translate original program's system calls into
- events
- Sort events by time and treat as independent
- Parallel program: events with conditionals specifying which node performs each event
- Ignore non-user I/O (such as loading libraries)

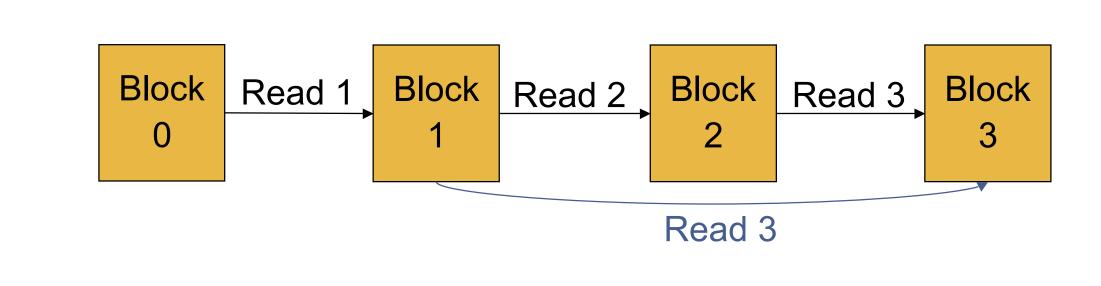


## Hidden Markov Model (HMM)

- HMM is a set of states with transition probabilities
- Some states are observable from data whereas others are hidden
- Applications in speech recognition and bioinformatics

### Hidden Markov Modeling of I/O Patterns<sup>1</sup> • Each file is broken up into segments (blocks)

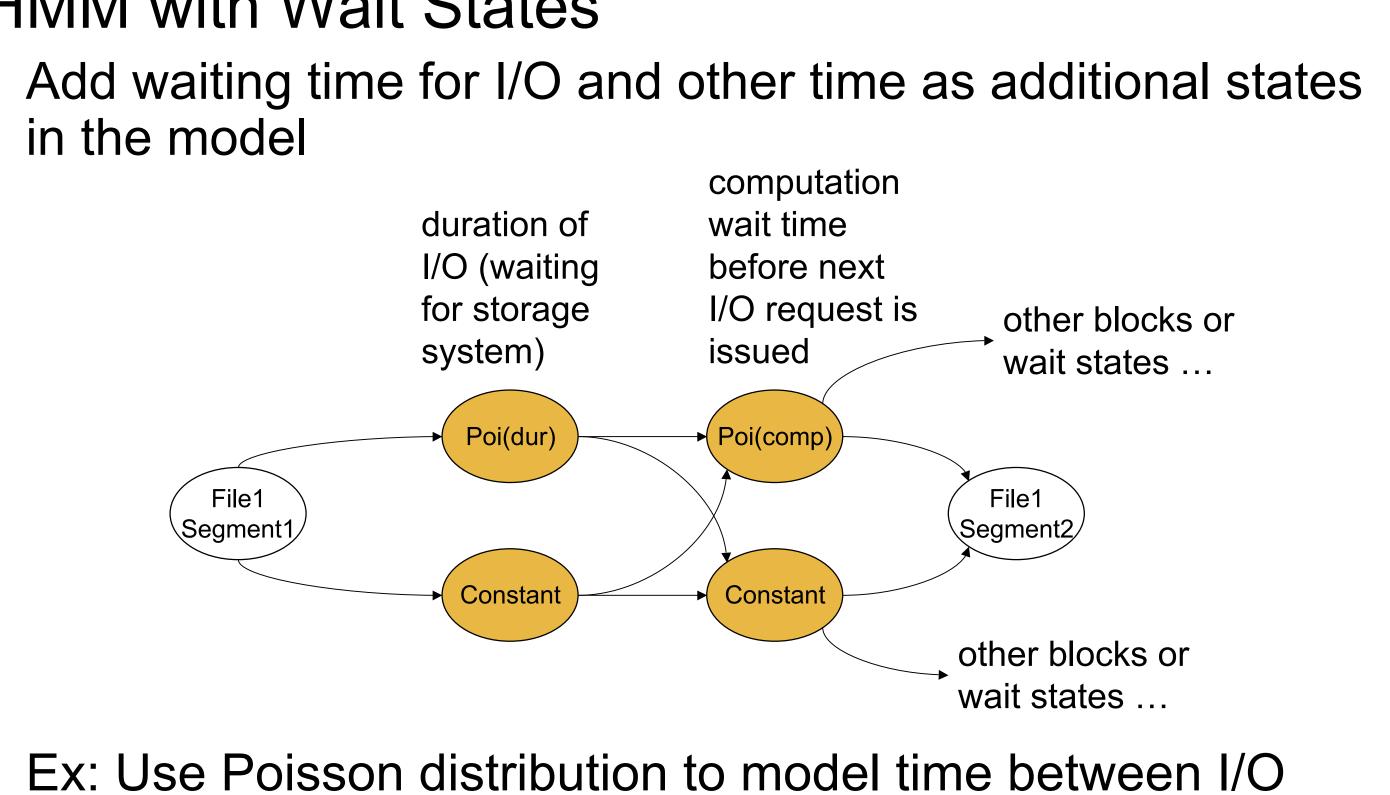
- Each segment is a state
- Transitions represent events, such as read of a single block
- Parallel access patterns require overlap in time and space
- Nodes should access similar file segments at similar times in order to be considered related
- Detects repeated accesses Useful for caching
- Determines workload changes dynamically
  - Useful for automatically changing storage policies



- Ex: HMM for sequential read of a file with possibility of skipping block 2
- <sup>1</sup> Tara M. Madhyastha and Daniel A. Reed. Input/output access pattern classification using hidden Markov models. In Proceedings of I/O in Parallel and Distributed Systems (IOPADS '97), pages 57-67, November 1997.

## HMM with Wait States

in the model



events' start times

### Prediction

- storage system
  - quickly, thus reducing I/O bottlenecks
- Predict future accesses within a workload
  - knowledge of prior accesses

### Conclusion

- Benchmarking a storage system requires great understanding of typical I/O workloads
- workloads to use as benchmarks

### Acknowledgements

- Funded by ISSDM and a GAANN fellowship
- **Alamos National Laboratory**



### Predict performance of this workload on a different

 Upgraded storage system might process I/O more Specify system characteristics in the workload - Use established I/O prediction techniques, based on

Using real I/O workloads can produce better synthetic

Thanks to Storage Systems Research Center and Los