



Fine-Grained Tracking of Grid Infections

Ashish Gehani

SRI

Basim Baig, Salman Mahmood, Dawood Tariq, Fareed Zaffar

LUMS





Introduction

- Grid semantics
 - *Not* middleware-specific
 - Distributed system
 - “Application community”
- Infection
 - **Security**, Reliability, Quality-of-Service
- Constraints
 - Fine-grained monitoring
 - Grid-wide correlation
 - Timely analysis

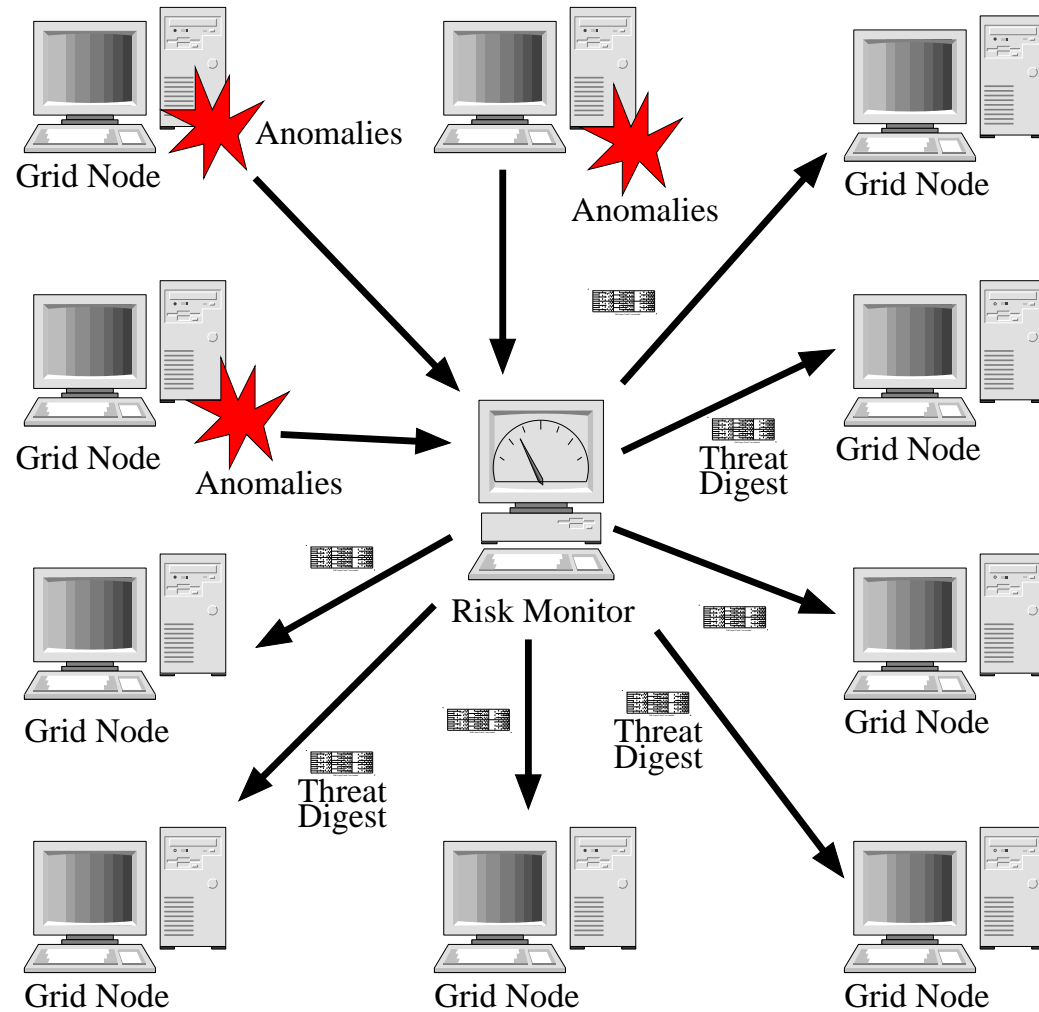




Motivation

- Attractive attack platform
 - Access to large set of resources
- Automatic privilege escalation
 - Single sign-on
- Significant consequences
 - Integrity loss of valuable data
- Exposed services
 - Open ports for callbacks

Application Community



Central Monitoring

- Collect Grid-wide anomalies
- Raw stream saturates network
- 35 clients, 10Mb/s (Oliner et al, RAID 2010)
 - Only event types
 - No arguments
- Must scale to hundreds of nodes



Local Monitoring

- Framed as set operations
- Application activity
 - Set of events
- Normal behavior
 - Union of events during training
- Anomalous behavior
 - Difference set (by subtracting normal)
- Correlating node activity
 - Intersection of anomaly sets

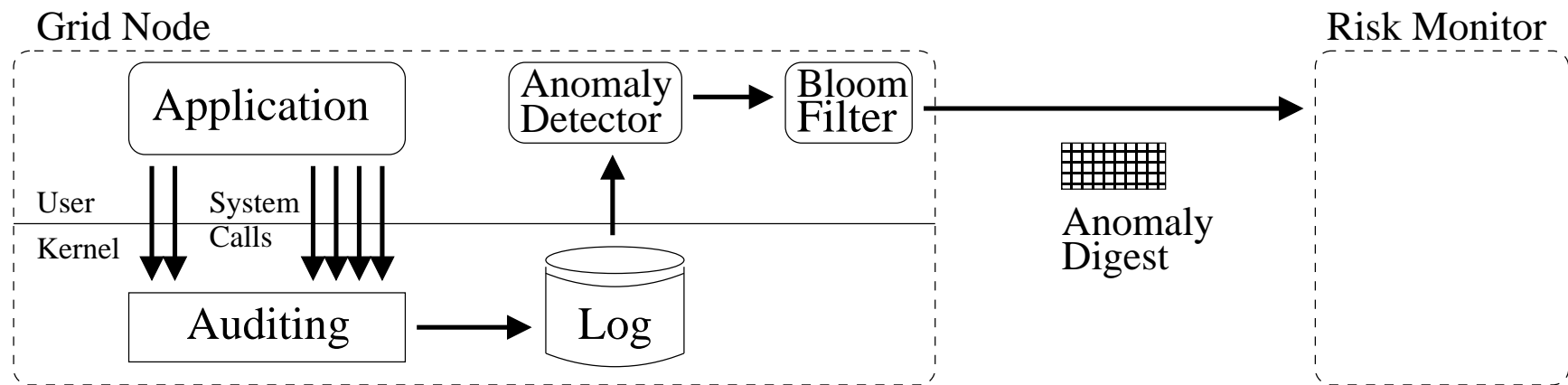




Approach

- Decompose sets into *epochs*
- Compress epoch activity
- Collect data provenance
- Map anomalies to provenance

Epoch Compression

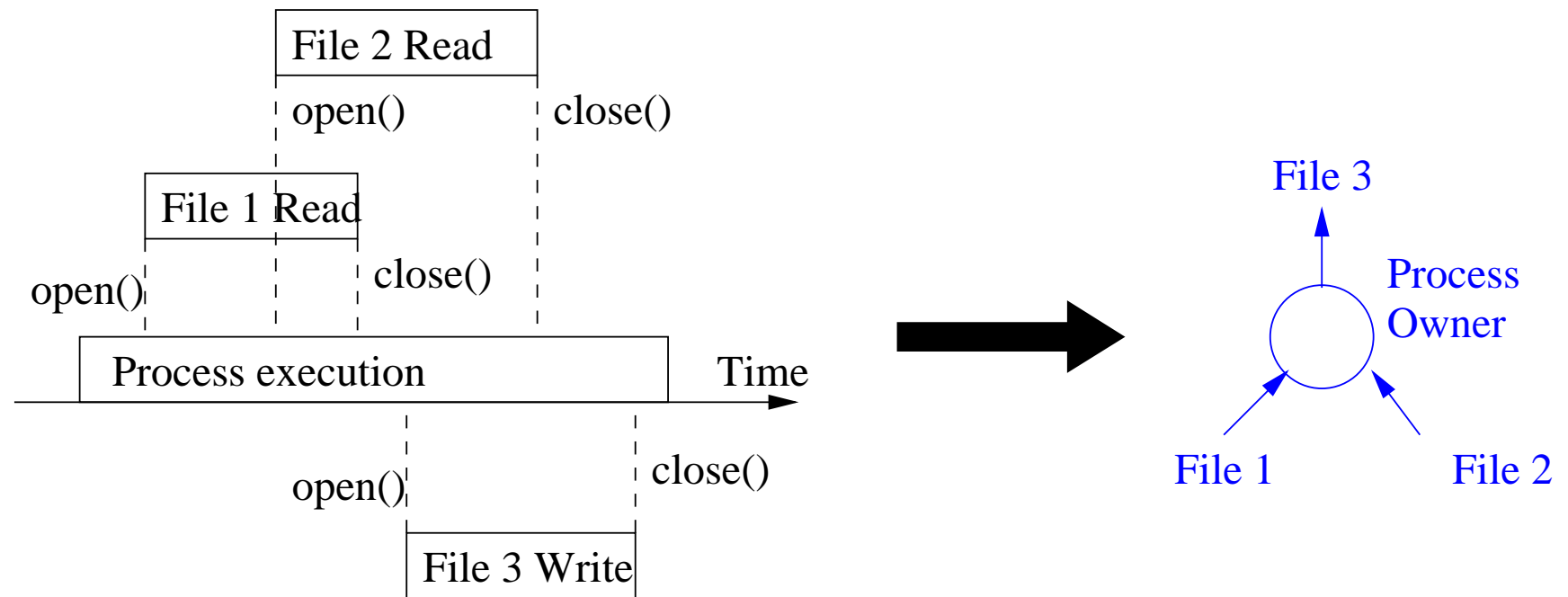


- Set representation
 - Allows use of Bloom filters
- Fold filter $\lceil \log(f + b) \rceil$ times
 - Increase update frequency by f
 - Decrease bandwidth used by b
 - More false positives

Correlating Activity

- Combine Bloom filters
 - Counting filter
- Event on τ nodes
 - Corresponding buckets are $\geq \tau$
- Construct *vaccination*
 - Bloom filter bit 1 \iff counting filter bucket $\geq \tau$

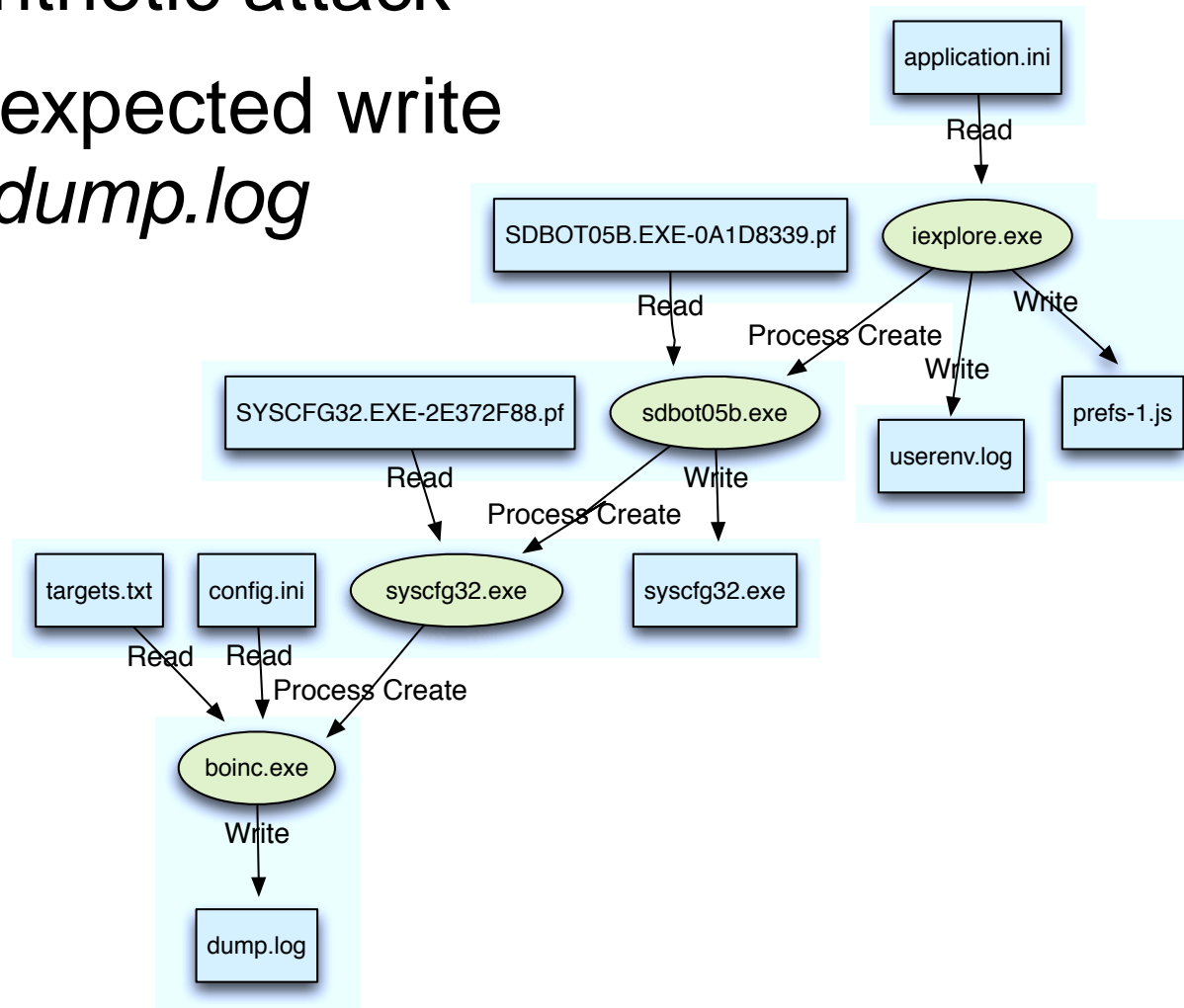
Data Provenance



- Record *few* arguments
 - Process creation, File versions
 - File access, modification

Anomaly Tracking

- Synthetic attack
- Unexpected write of *dump.log*





Evaluation Platform

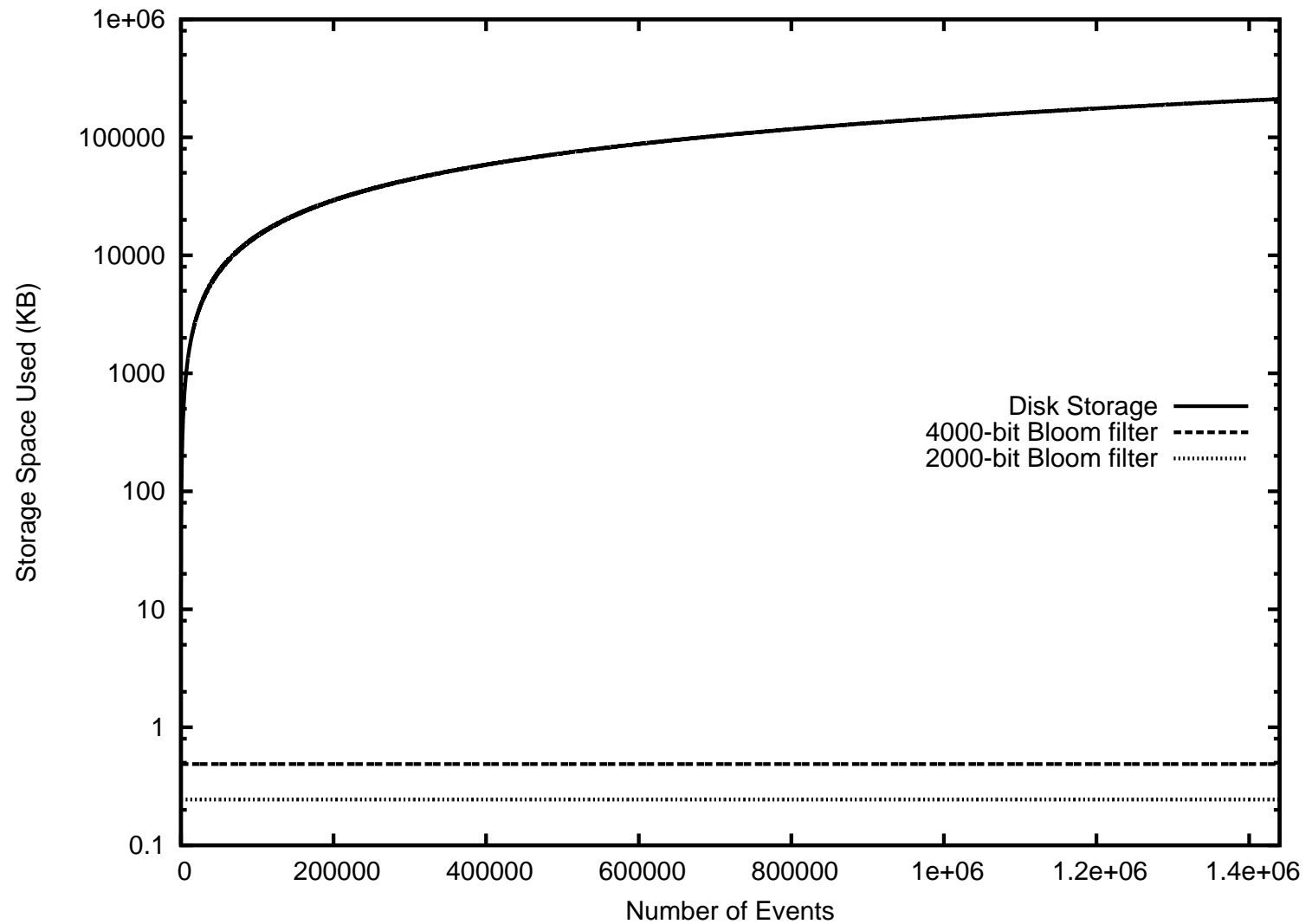
- Microsoft Windows XP (SP3)
- BOINC 6.10.43 volunteer Grid application
- Process Monitor 2.7 tool
- Open Bloom Filter library
- Synthetic infection
 - Internet Explorer vulnerability
 - Windows *CreateRemoteThread()*
 - MailBoy 2004 injected

Workload

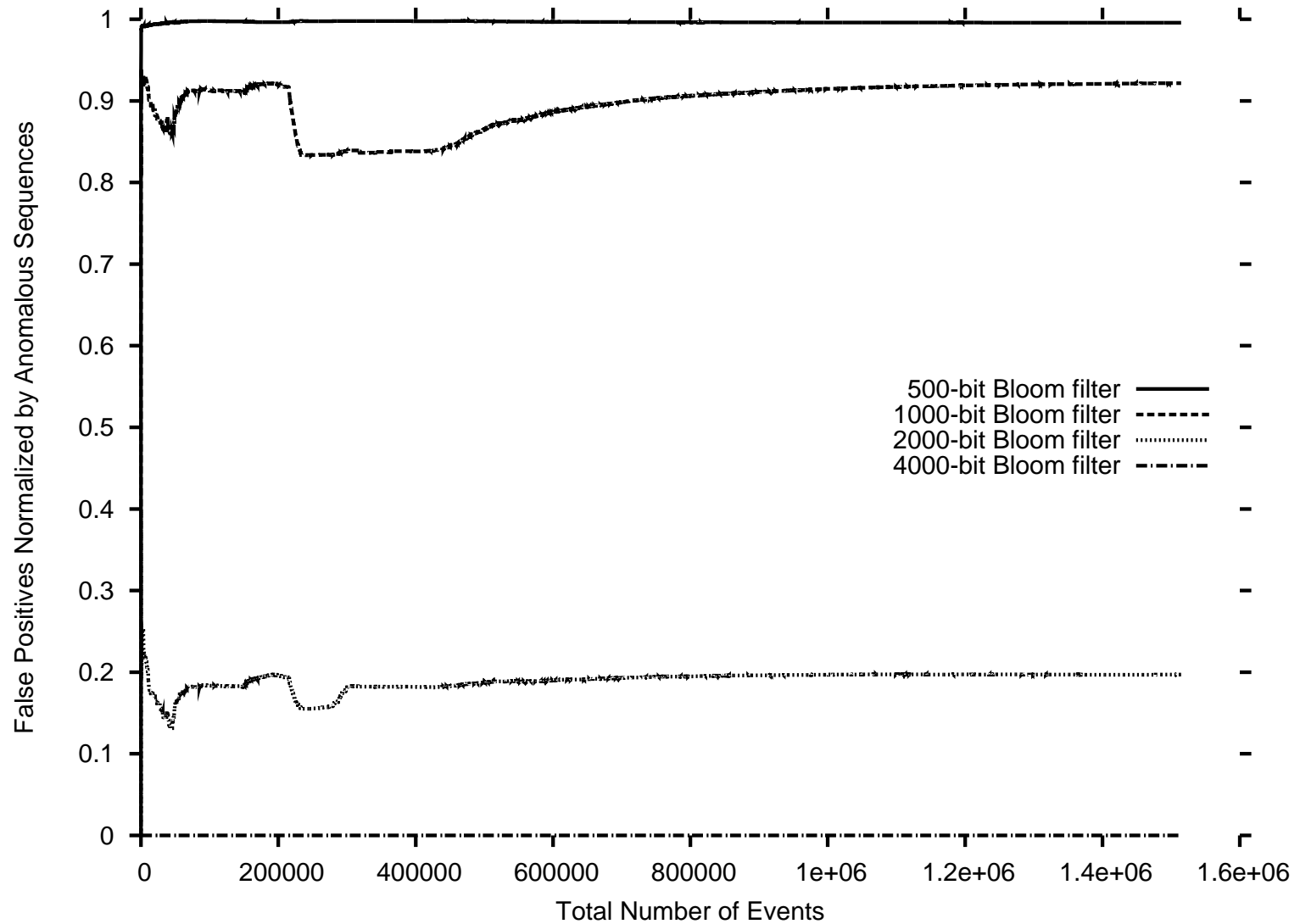
- 24 hours 20 minutes
- 1.5 million events
- Raw log: 216 MB / Grid node
- Anomaly detection with 11-tuples
- MailBoy 2004 as spam relay
 - 20 threads
 - 30 second timeout
 - 1,700 email addresses



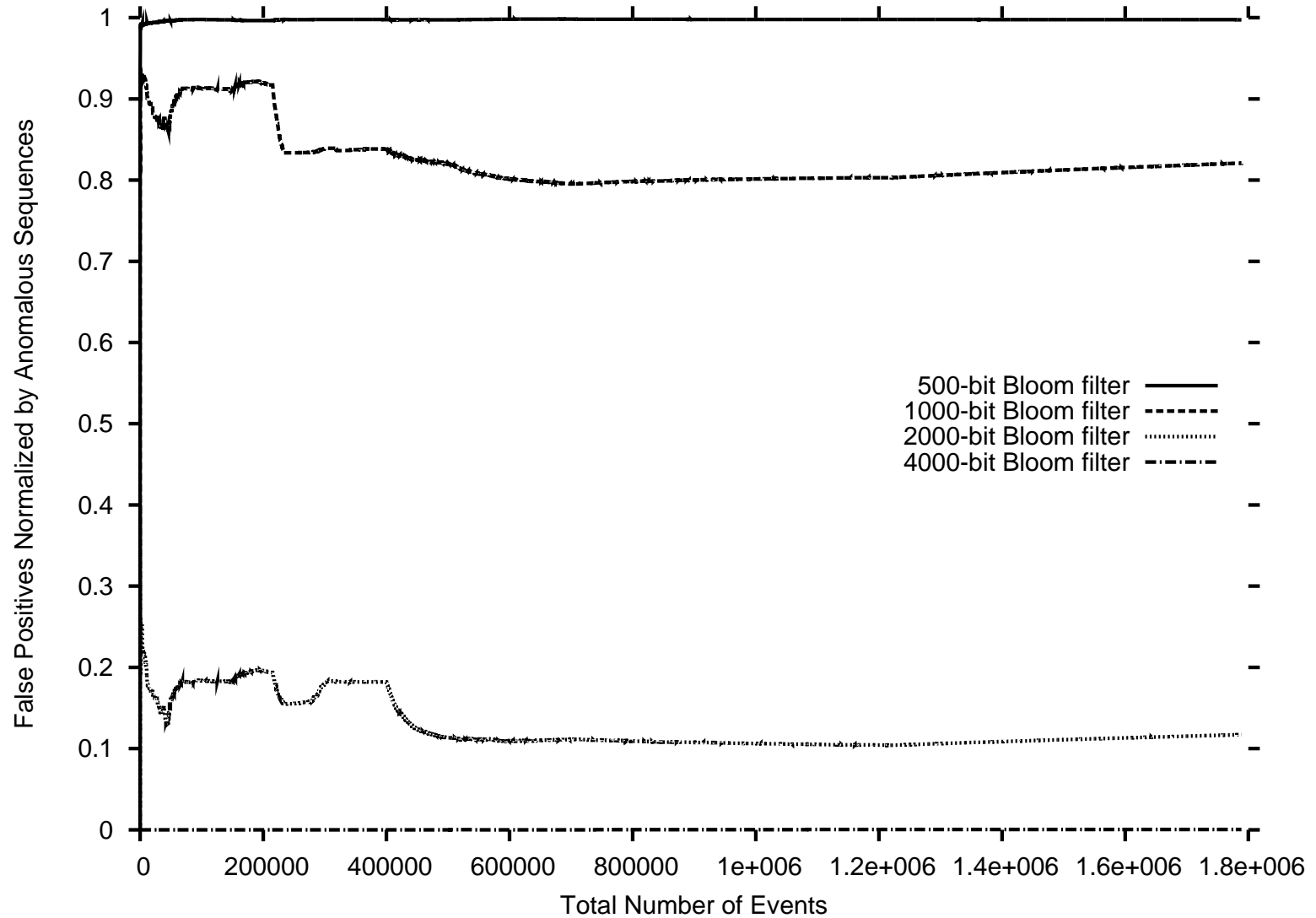
Storage



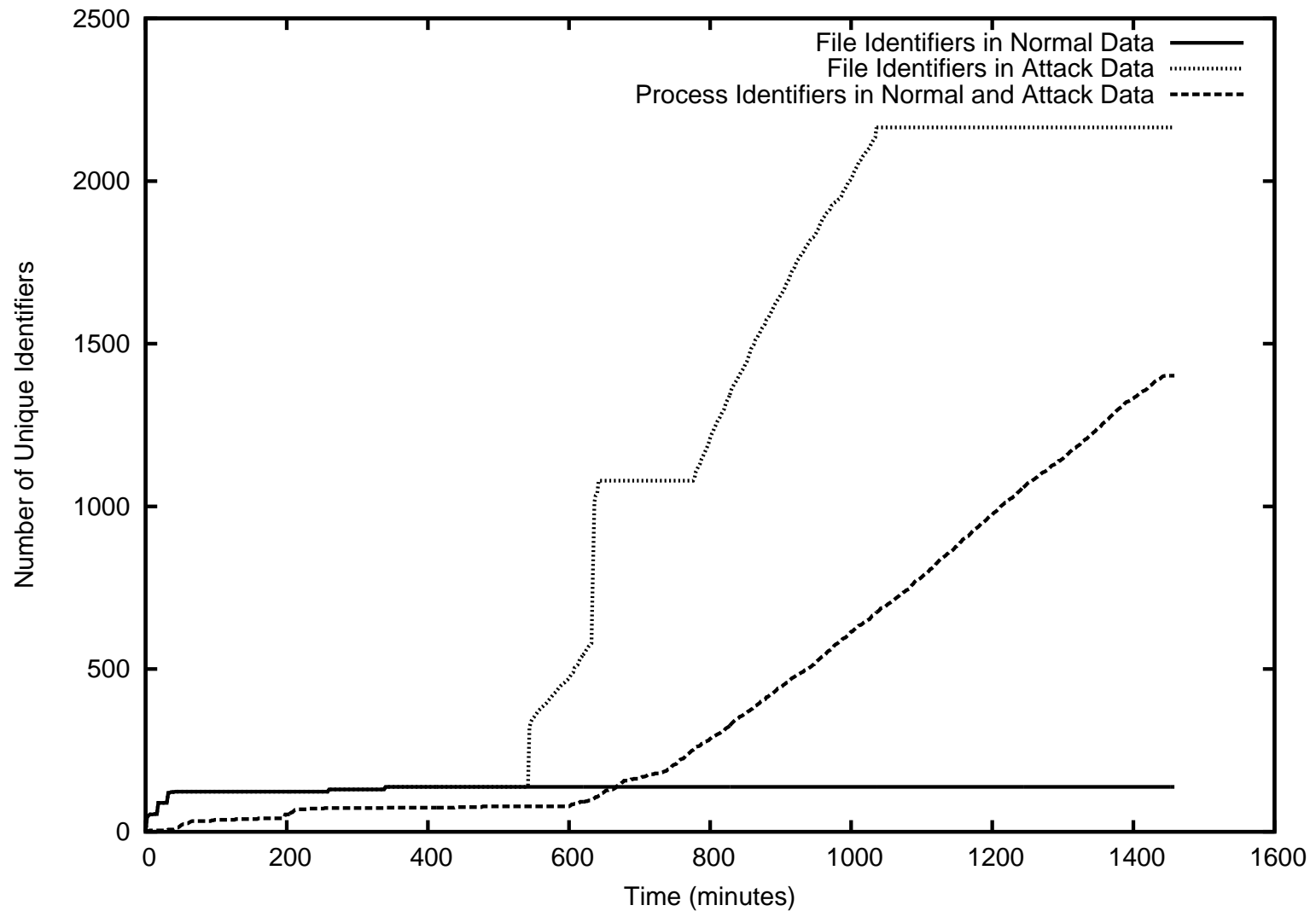
Normal Operation



Malware Injected



Provenance Database





Conclusion

- Apparent tension
 - Fine-grained anomaly detection
 - Grid-wide monitoring
- Solution
 - Audit provenance on Grid nodes
 - Compress event stream
 - Map anomalies to provenance
- Acknowledgement
 - NSF Grant OCI-0722068

