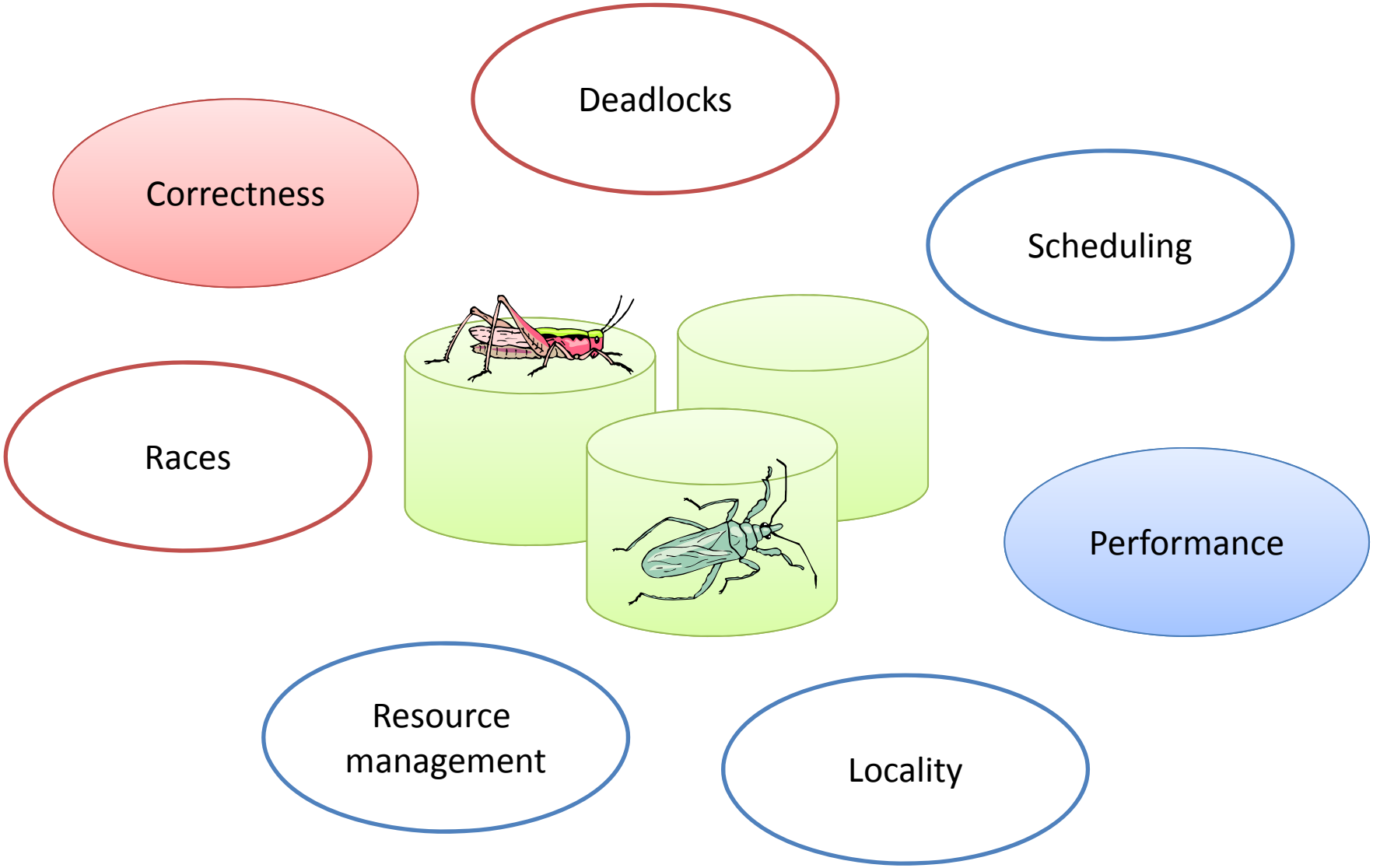
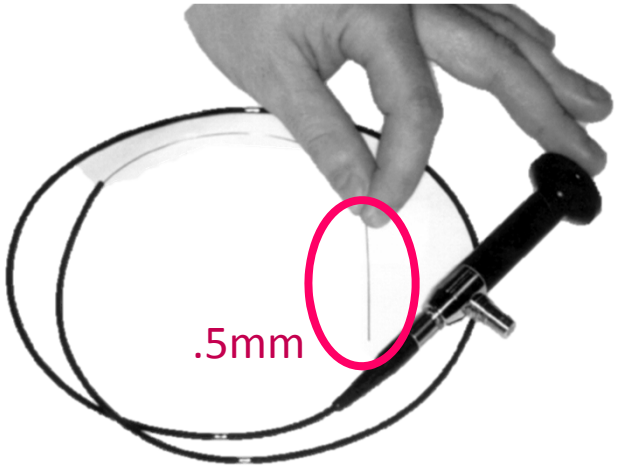
Fighting back: Using observability tools to improve the DBMS (not just diagnose it)

Ryan Johnson

DBMS/OS interface is tricky terrain



Fine-grained, targeted observation?



+

Image credit: Müllner, Bodner, Mannor. BJO 83:8, 1999.

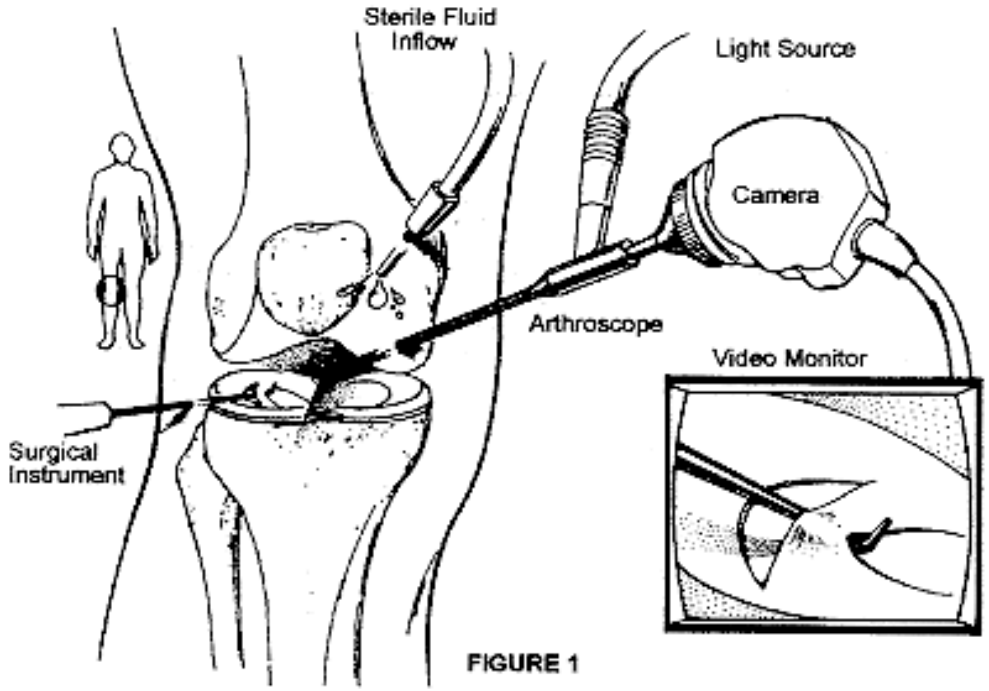


Image credit: Johns Hopkins Dept. of Orthopedic Surgery

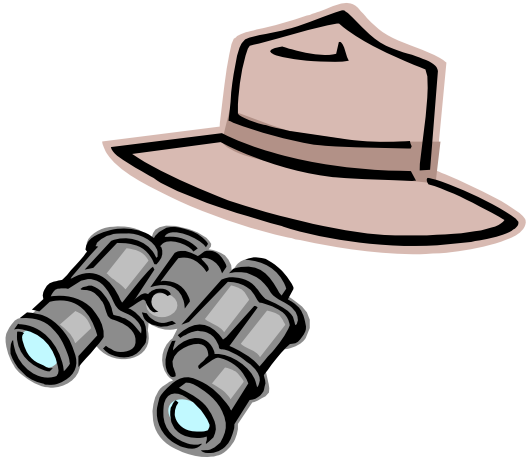
Real goal: fine-grained, targeted fixes

DTrace: making systems transparent

Instrument majority of OS and all user-level functions

Zero disabled probe cost

Trace events generated by domain-specific "providers"



Safe: designed for non-wizard use with production systems

Domain-specific language specifies probe actions

Two-way OS/app communication via process memory

Disclaimer

This talk is about a new way of looking at our tools, not about any one (use of a) tool.

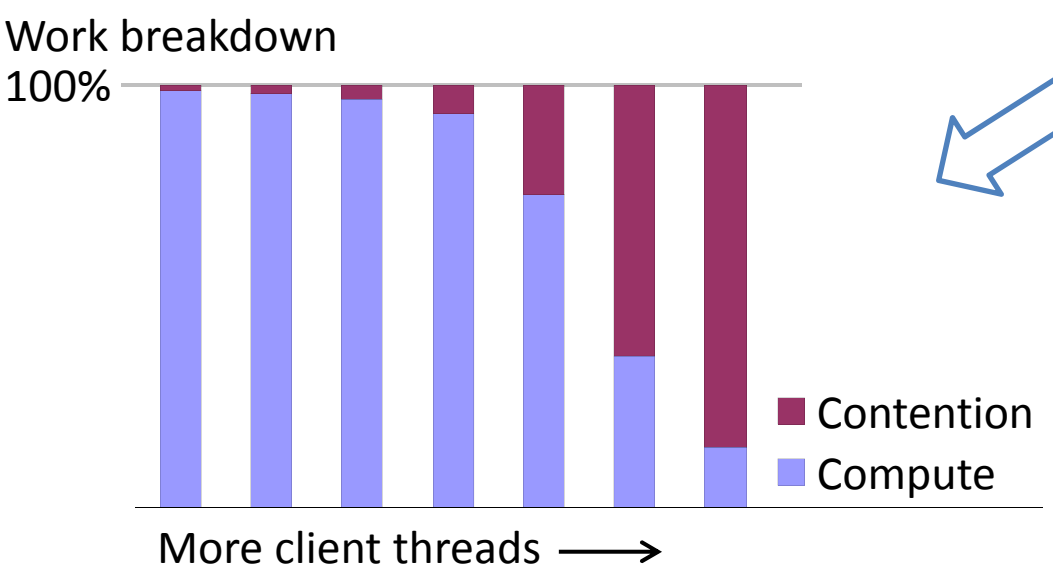
For concreteness, we'll discuss a specific tool (DTrace) and problem class (scheduling) that I happen to be familiar with.

Hopefully these examples inspire other/better uses for active observers within the DBMS.

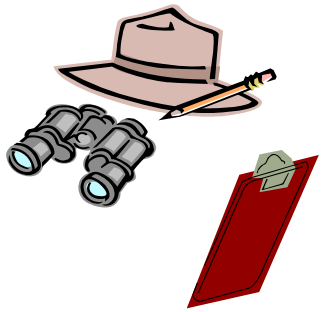
Example: diagnosing contention

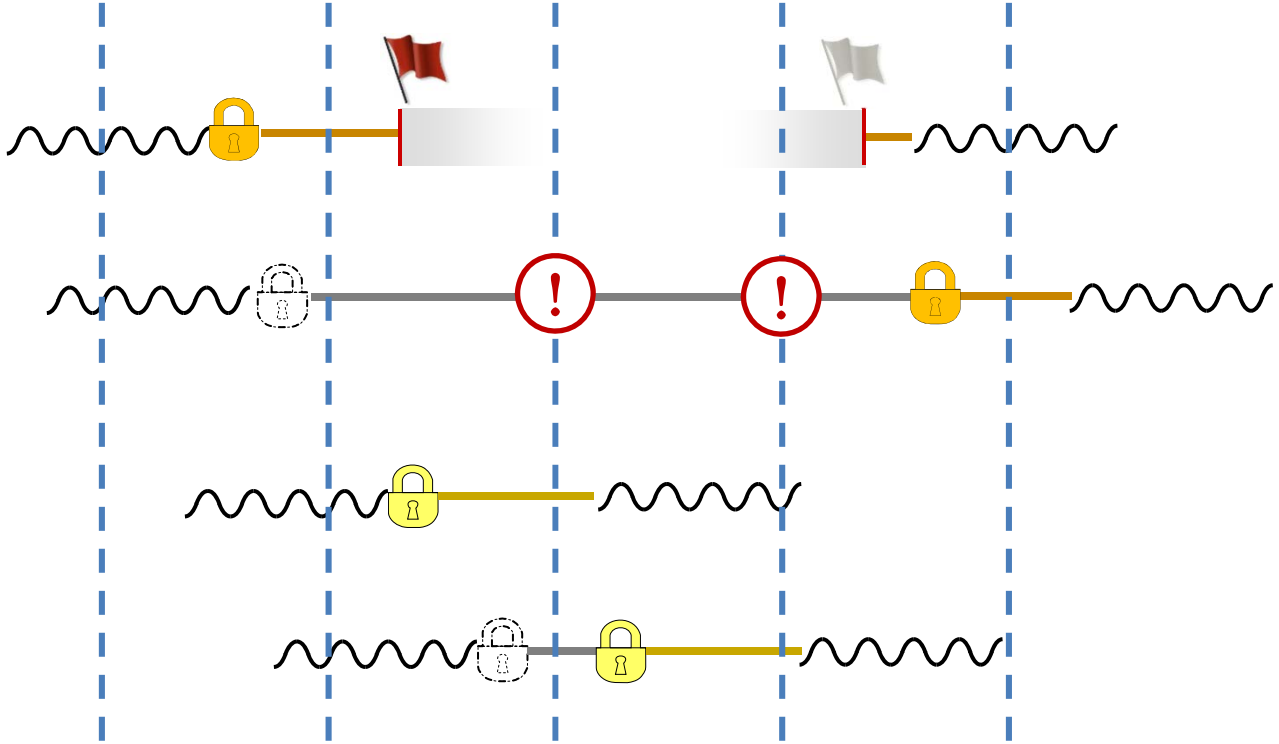
Sample-based profiling (OS service)

Post-processing identifies samples due to contention (lock waits, atomic ops, etc.)



Spinning and thread preemption

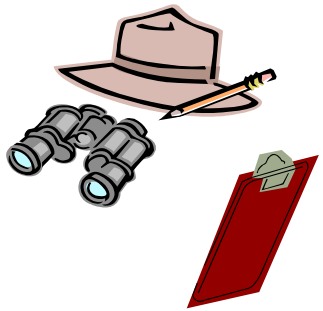




DBMS records lock acquire/release, DTrace does the rest

Priority inversion unmasked

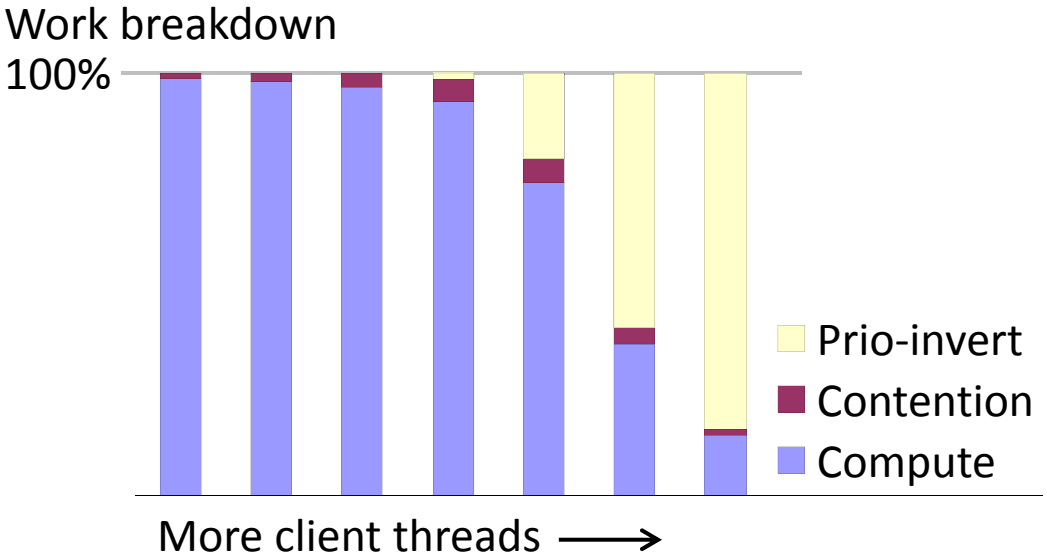
DTrace tracks locks held by pre-empted threads, samples call stacks



DTrace isolates samples where threads wait on pre-empted locks

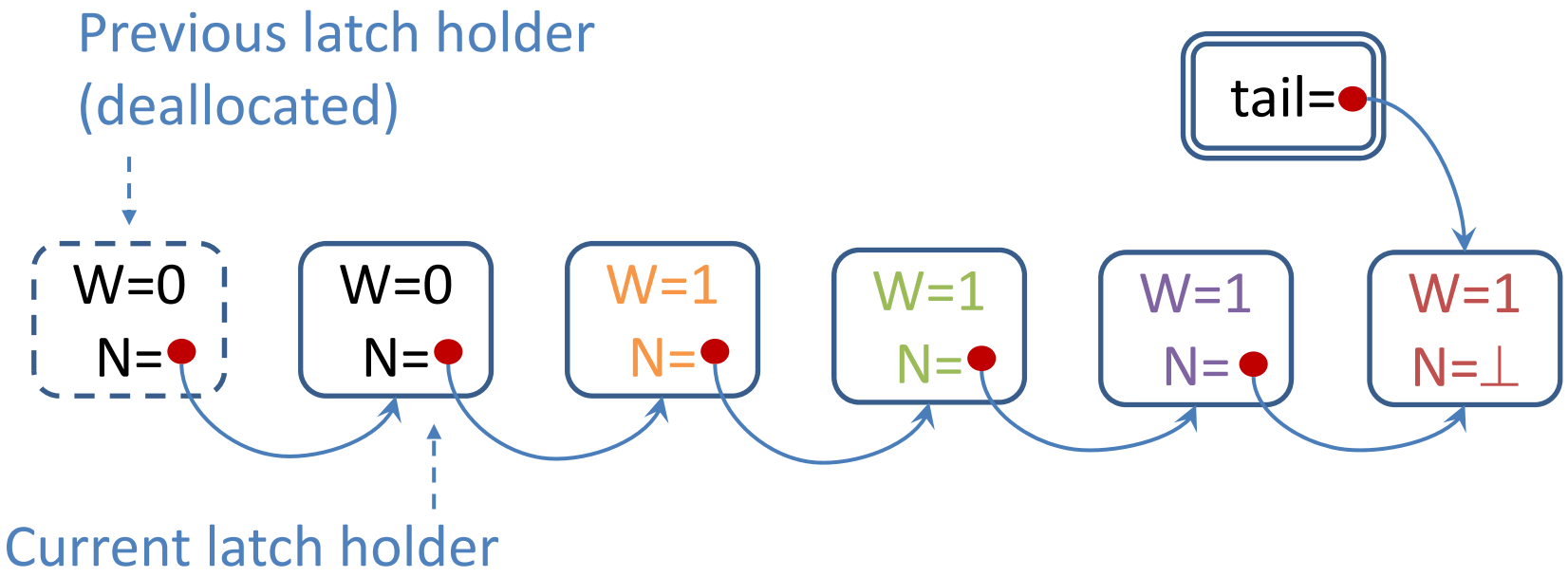
Post-processing differentiates samples due to contention and priority inversion

Worker threads record held/attempted locks in thread-local storage

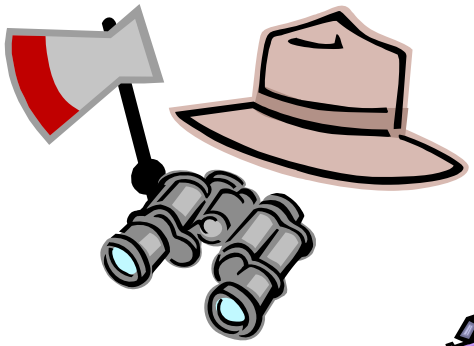


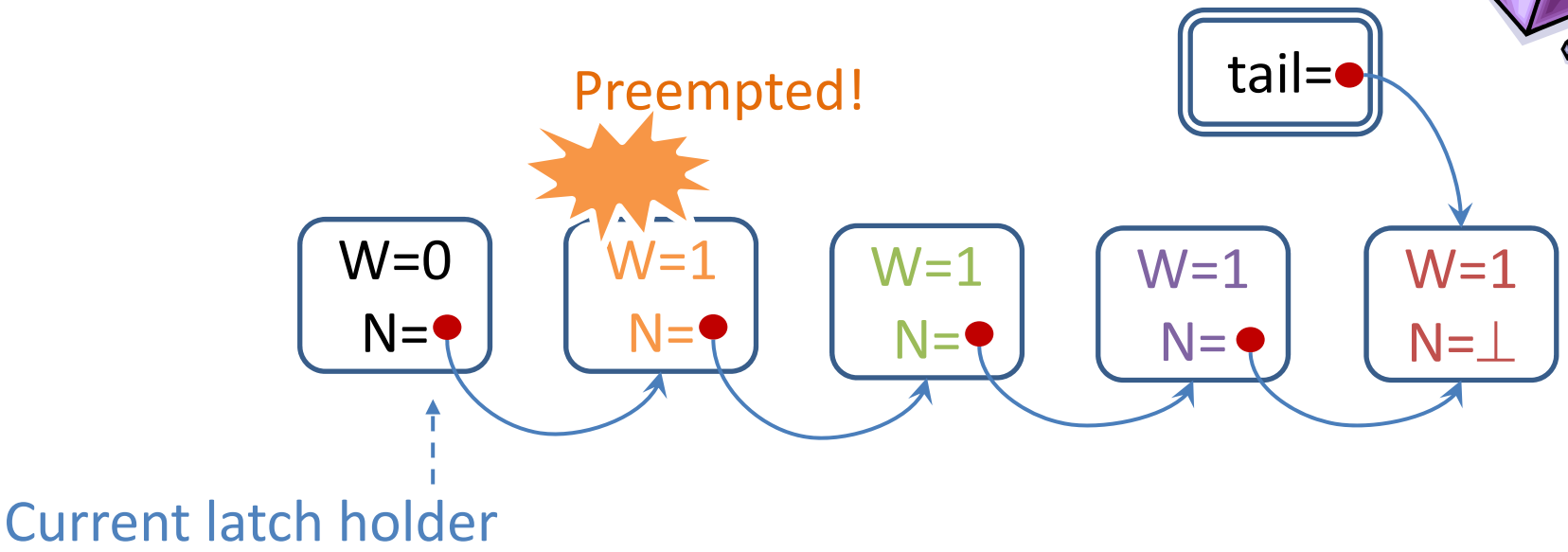
Neither OS nor DBMS can achieve this alone

Example: Queuing and pre-emption

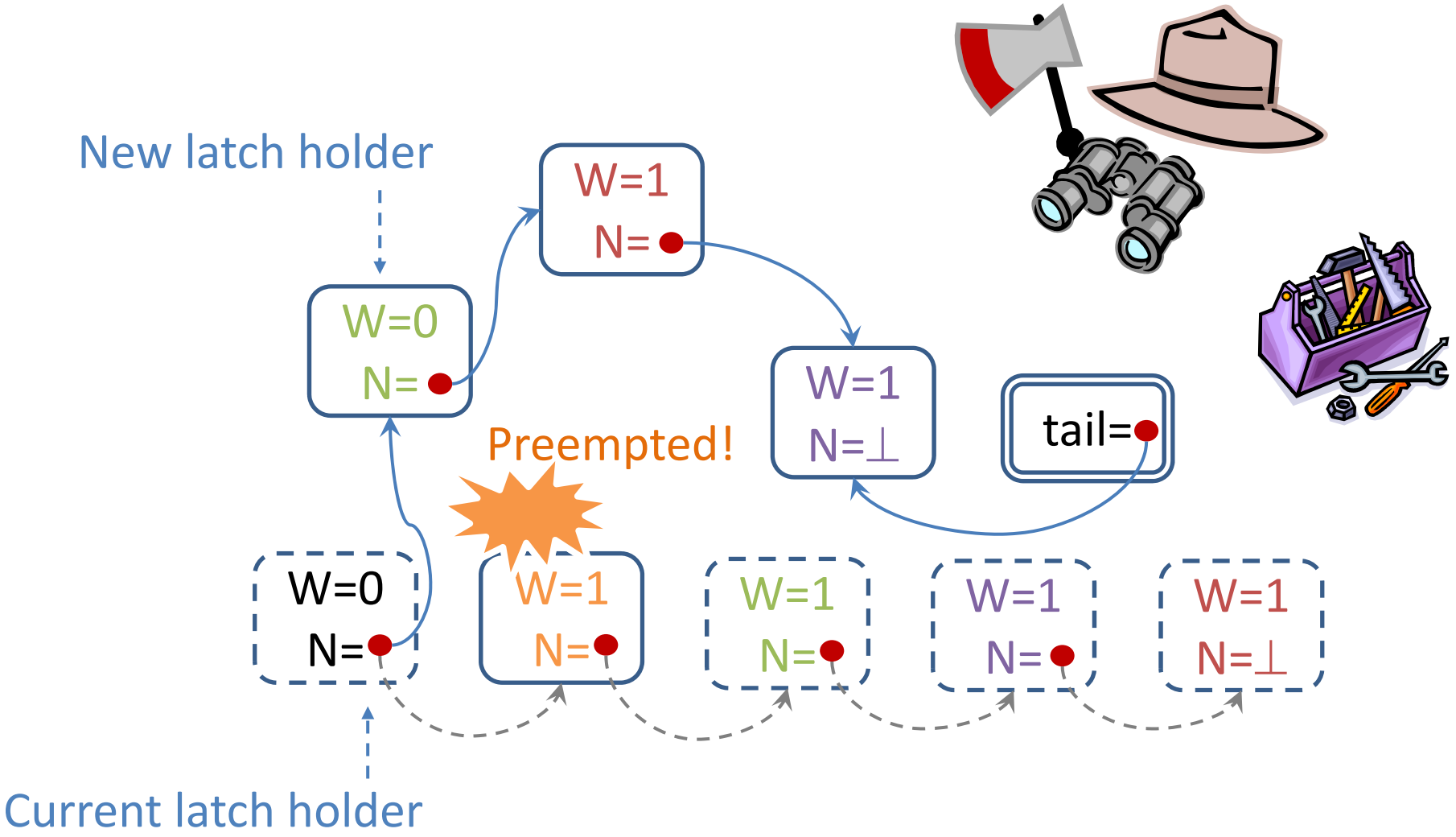


Example: Queuing and pre-emption





Preemption-resistant queuing



DTrace dissolves queue, wakeful threads simply rejoin

Conclusions

- Cooperation between DBMS and observability tools can be a powerful thing
- Recurring theme: two-way communication between OS/DBMS
- What other active uses might be out there?
- How to adapt our tools for these new uses?