Concurrency Control and Performance

Agrawal/Carey/Livny: Locking vs. Optimistic

Previous work had conflicting results:
• Carey & Stonebraker (VLDB84), Agrawal & DeWitt (TODS85): blocking beats restarts
• Tay (Harvard PhD) & Balter (PODC82): restarts beat blocking
• Franaszek & Robinson (TODS85): optimistic beats locking

Goal of this paper:
• Do a good job modeling the problem and its variants
• Capture causes of previous conflicting results
• Make recommendations based on variables of problem

Methodology
• simulation study, compare Blocking (i.e. 2PL), Immediate Restart (restart when denied a lock), and Optimistic (a la Kung & Robinson)
• pay attention to model of system:
  • database system model: hardware and software model (CPUs, disks, size & granule of DB, load control mechanism, CC algorithm
  • user model: arrival of user tasks, nature of tasks (e.g. batch vs. interactive)
  • transaction model: logical reference string (i.e. CC schedule), physical reference string (i.e. disk block requests, CPU processing bursts).
• Probabilistic modeling of each. They argue this is key to a performance study of a DBMS.
• logical queueing model
• physical queueing model

Measurements
• measure throughput, mostly
• pay attention to variance of response time, too
• pick a DB size so that there are noticeable conflicts (else you get comparable performance)

Experiment 1: Infinite Resources
• as many disks and CPUs as you want
• blocking thrashes due to transactions blocking numerous times
• restart plateaus: adaptive wait period (avg response time) before restart
  • serves as a primitive load control!
• optimistic scales logarithmically
• standard deviation of response time under locking much lower

Experiment 2: Limited Resources (1 CPU, 2 disks)
• Everybody thrashes
• blocking throughput peaks at mpl 25
• optimistic peaks at 10
• restart peaks at 10, plateaus at 50 – as good or better than optimistic
• at super-high mpl (200), restart beats both blocking and optimistic
• but total throughput worse than blocking @ mpl 25
• effectively, restart is achieving mpl 60
• load control is the answer here – adding it to blocking & optimistic makes them handle higher mpls better

Experiment 3: Multiple Resources (5, 10, 25, 50 CPUs, 2 disks each)
• optimistic starts to win at 25 CPUs
  • when useful disk utilization is only about 30%, system begins to behave like infinite resources
• even better at 50

Experiment 4: Interactive Workloads
Add user think time.
• makes the system appear to have more resources
• so optimistic wins with think times 5 & 10 secs. Blocking still wins for 1 second think time.

Questioning 2 assumptions:
• fake restart – biases for optimistic
  • fake restarts result in less conflict.
  • cost of conflict in optimistic is higher
  • issue of k > 2 transactions contending for one item
  • will have to punish k-1 of them with real restart
• write-lock acquisition
  • recall our discussion of lock upgrades and deadlock
  • blind write biases for restart (optimistic not an issue here), particularly with infinite resources
    (blocking holds write locks for a long time; waste of deadlock restart not an issue here).
  • with finite resources, blind write restarts transactions earlier (making restart look better)

Conclusions
• blocking beats restarting, unless resource utilization is low
• possible in situations of high think time
• mpl control important. admission control the typical scheme.
  • Restart’s adaptive load control is too clumsy, though.
• false assumptions made blocking look relatively worse

Final quote by Wulf!