Caching

Rob Abbott
CMPS 242
Fall 2009

Based on work by Robert Gramacy
Thanks to Prof. Warmuth for providing the problem and Gramacy's data.
The Problem

• Want to design a good caching scheme
  • Using Experts
    – Each expert has a virtual cache
    – Real cache policy relies on experts to make choices
    – Window based algorithms?
    – Weighted Majority with share updates?

• Want to find a good comparator
Window vs Weighted Majority

- Window puts best expert (of P expert policies) over past N trials (N=window size) in charge of the real cache
- Weighted Majority uses weighted votes to manage the real cache

- Window requires $O(N^*P)$ additional space
- Weighted Majority generally requires $O(P)$ additional space
- Window algorithms can be thought of as weighted majority with a binary weight update
FSUP(\eta = 0.25) Weights
Window Algorithm (size = 1000)
Comparators

- **AllVC**
  - Uses all virtual caches
  - Will have 0 unforced misses if sufficiently many/varied experts are used

- **Alternative**
  - Cache Entry Priority = \((\text{time till next occurrence})^a \cdot \text{size}^b\)
  - Uses offline future knowledge
  - Otherwise plays by same rules as other policies
  - Provably optimal when cache entries are of fixed size
Conclusions

- Alternative Comparators are very good
  - They don't cheat
  - Sometimes beat All-VC
  - Priority = (time till next) * (size)^2 did well consistently

- Window algorithms aren't great
  - They tend to only do well when they track the best fixed expert

- Adjusting virtual cache sizes doesn't help if the virtual caches occupy enough real cache space to hurt the master algorithm and doesn't help much otherwise

- Gramacy's main point was that refetching to the ideal cache helps tremendously. I don't address that here.