Contribution ID: **176**   Type: **not specified**

Sharing Cache - optimizing for a single core vs a multi-core system

Wednesday, 22 September 2021 08:15 (15 minutes)

Recent x86 processors support "non_temporal" stores which bypass the cache when storing data. It is widely understood that normal stores to cache are appropriate when it is likely that the data may be needed before the cache is full. It is also understood that stores of large blocks of data which exceed the available cache allow the overall application to run faster when the block of stores bypass the cache, leaving other locally used data in the cache. A recent change (since reverted) tuned the library routine for memcpy to optimize based on best results assuming a single core was the sole user of the cache instead of allowing for multi-core server chips which have multiple cores sharing a chip. The specifics of the two cases will be presented followed by discussion of how similar single core vs multi-core optimizations might be handled in standard software libraries.

I agree to abide by the anti-harassment policy

I agree

Primary author: MCGEHEARTY, Patrick (Oracle)

Presenter: MCGEHEARTY, Patrick (Oracle)

Session Classification: GNU Tools Track

Track Classification: GNU Tools Track