

Data Caching: Data Consistency

Instructor: Dr. Sunho Lim (Ph.D., Assistant Professor)

Lecture 03

sunho.lim@ttu.edu

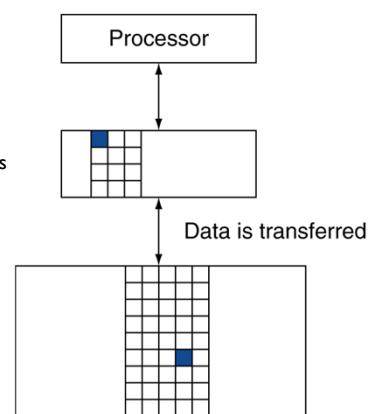
CS5331: Mobile Data Management and Privacy, Spring 2023



1

Cache: Memory Hierarchy

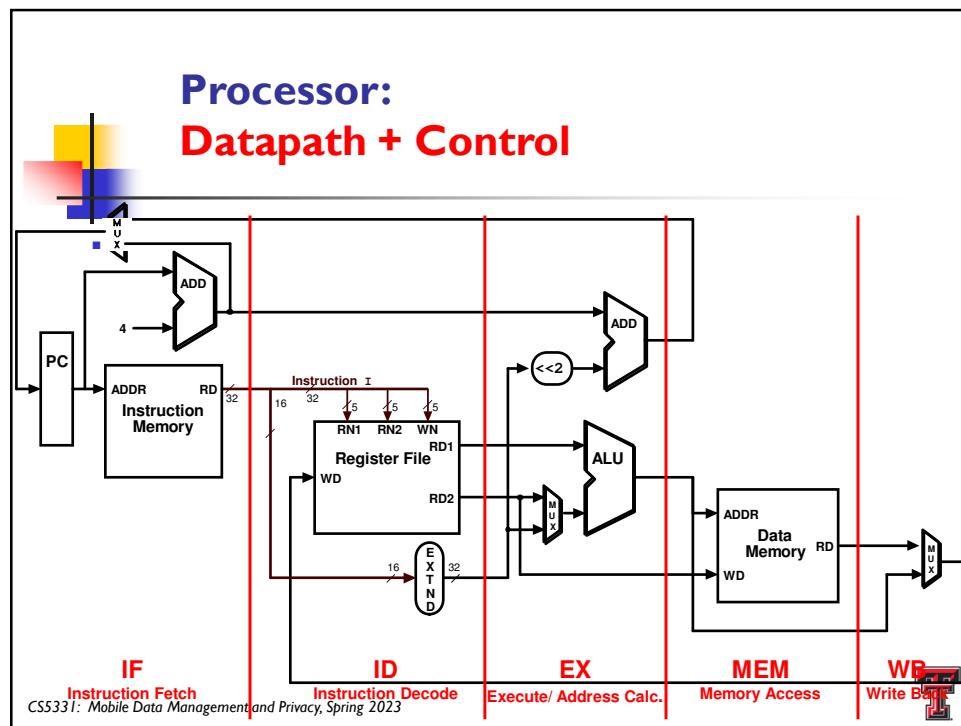
- Located between the CPU and main memory
- Memory hierarchy:
 - A level closer to the processor is a **subset** of any level further away
 - All the data is stored at the lowest level
 - Data is copied between only two adjacent levels at a time
 - Upper level:
 - Faster and smaller
 - Lower level:
 - Slower and bigger



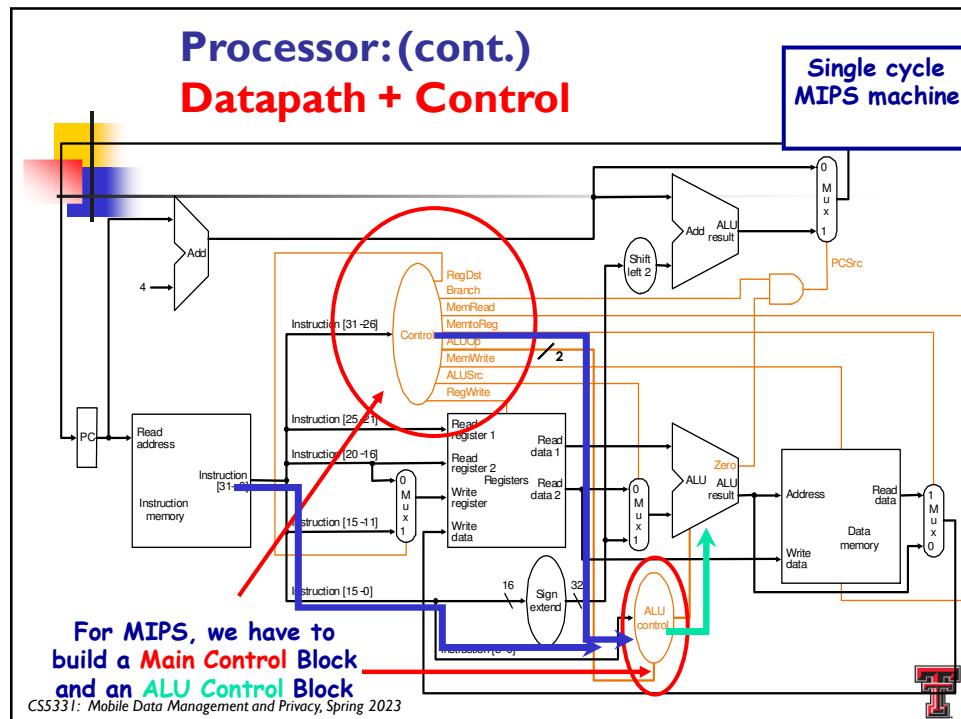
CS5331: Mobile Data Management and Privacy, Spring 2023



2



3



4

Cache: Locality

for ($i = \phi; i < 10; i++$)
 $s = h + A[i];$

- Principle of locality:
 - Programs access a relatively **small portion** of their address space at any instant of time
 - **Temporal locality:**
 - Locality in time
 - If an item is referenced, it will tend to be referenced **again** soon
 - **Spatial locality:**
 - Locality in space
 - If an item is referenced, items whose addresses are **close** by will tend to be referenced soon
- Almost every general-purpose machine includes a cache

CS5331: Mobile Data Management and Privacy, Spring 2023

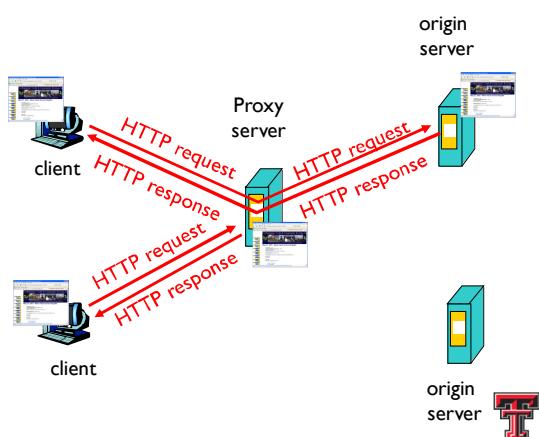


5

Web Caches (Proxy Server)

Goal: satisfy client request without involving origin server

- user sets browser: Web accesses via cache
- browser sends all HTTP requests to cache
 - object in cache: cache returns object
 - else cache requests object from **origin server**, then returns object to client



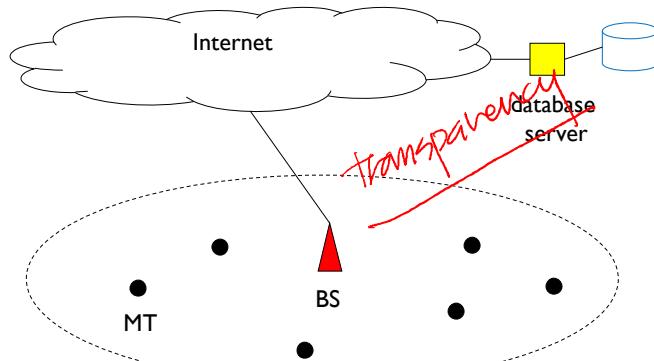
CS5331: Mobile Data Management and Privacy, Spring 2023



6

System Model: Cellular Network

→ header



CS5331: Mobile Data Management and Privacy, Spring 2023



7

Most Caching Techniques

- How to fulfil the users' insatiable interests in accessing the Internet wirelessly?
 - One of key optimization techniques is **Caching!!**
- Whether to maintain the cache status or not
 - Stateful and Stateless approaches
- Who initiate cache validity
 - Push and Pull approaches
- **Level of cache consistency**
 - Strong, Weak, and Hybrid approaches

CS5331: Mobile Data Management and Privacy, Spring 2023



8

Most Caching Techniques: Data Consistency

- Strong consistency
 - Invalidation Report (IR)-based approach
 - Updated IR (UIR)-based approach
 - High consistency maintenance cost
- Weak consistency
 - Time-To-Live (TTL)-based approach
 - Predicting TTL value is not trivial
- Hybrid consistency
 - Probabilistic delta consistency approach
 - Predicting global update information is difficult

CS5331: Mobile Data Management and Privacy, Spring 2023



9

IR-based System Model

- **Invalid Report (IR) based:** Periodic broadcast an IR from the server
 - IR $[d^s, t^s]$: an id of updated data item and its most recent timestamp, respectively
 - Update history during the last $L * |W_b|$

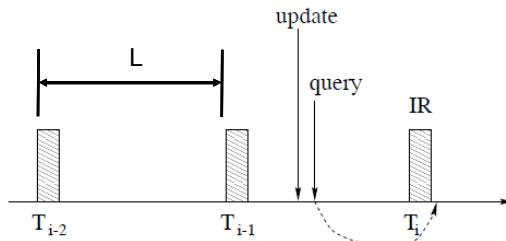


Figure 1: The IR-based cache invalidation model

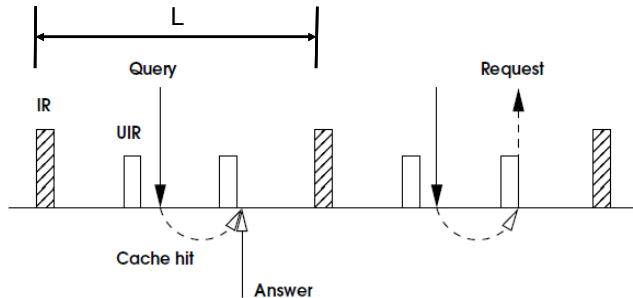
CS5331: Mobile Data Management and Privacy, Spring 2023



10

UIR-based System Model

- **Update IR-based:** A set of UIRs broadcasted between IRs



CS5331: Mobile Data Management and Privacy, Spring 2023



11

Motivation

- In most caching schemes,
 - A **strong consistency** is implicitly assumed
 - A query is answered by the latest updated data item from either a server or local cache
- Ensuring a strong consistency requires a non-negligible long query delay
 - Confirm both source data item and its cached copy are consistent
- Note that ensuring a strong consistency is not always a prompt and critical requirement!!
 - A consistency requirement may be different depending on an update sensitivity

CS5331: Mobile Data Management and Privacy, Spring 2023



12

Motivation (cont.)

- For example, update insensitive information
 - Maps, video clips, e-flyer, and weather information
 - Occasional inconsistency between the source data item and its cached copy would be acceptable



13

ConSens: Consistency-Sensitive Opportunistic Data Access

- A User-defined Cache Consistency
 - Each user can set its own consistency level
 - Maintain the consistency probabilistically
- Lazy Request and Opportunistic Data Access
 - Balance the data accessibility and query latency
 - Each user can judiciously access the cached data items
- Integrated the proposed technique into the existing IR- and UIR-based caching techniques

CS5331: Mobile Data Management and Privacy, Spring 2023



14

ConSens: Consistency Condition

- Each node can set its own consistency level
 - Set its own consistency level
 - **Target Consistency** (τ , $0 < \tau \leq 1$)
 - $\tau = 1$ implies a strong consistency
 - Maintain a number of invalidate data items accessed ($N^{invalid}$) during **ConsistencyWindow** (w_c)
 - Maintain **Current Consistency** (τ^{cur})
 - $\tau^{cur} = 1 - \frac{\sum_{u=0}^{|w_c|} Invalid[u]}{|w_c|}$
 - $Invalid[u]$: marking (i.e., I) for answering a query with an invalid data item
 - Examine **Consistency Condition**
 - $\tau \leq \tau^{cur}$

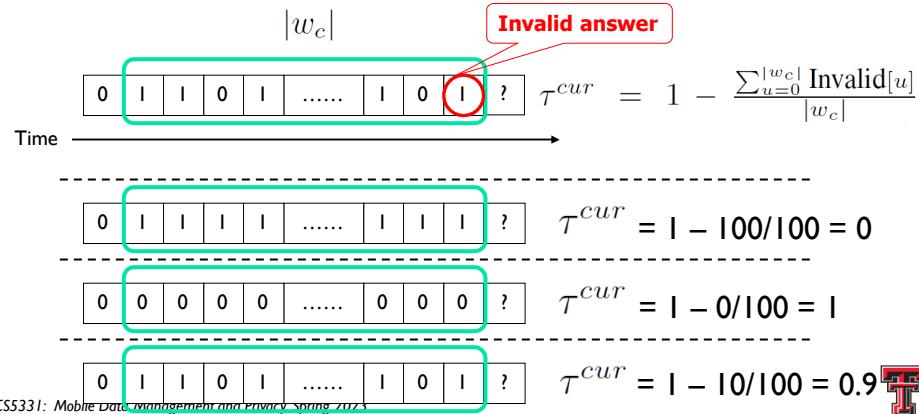
CS5331: Mobile Data Management and Privacy, Spring 2023



15

ConSens: Consistency Condition (cont.)

- For example,
 - Only consider the last $|w_c|$ number of queried data items



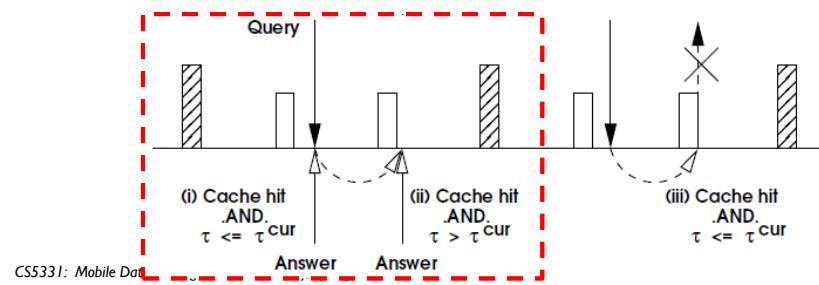
CS5331: Mobile Data Management and Privacy, Spring 2023



16

ConSens: Opportunistic Data Access

- When a query is generated which can be answered by a cached copy,
 - Check the Consistency Condition ($\tau \leq \tau^{cur}$)
 - If satisfied,
 - Immediately use the cached copy for answering the query without waiting for the next IR/UIR
 - If not satisfied, wait for the next IR/UIR



17

ConSens: Opportunistic Data Access (cont.)

- Accessing a cached copy without verifying it with the server, as far as
 - Consistency Condition ($\tau \leq \tau^{cur}$) is satisfied
 - The cached copy is marked as valid
- Recalculate Current Consistency (τ^{cur})
 - Wait for the next IR/UIR and check whether the cached copy was valid
 - Update Invalid[u] and recalculate τ^{cur}

$$\tau^{cur} = 1 - \frac{\sum_{u=0}^{|w_c|} \text{Invalid}[u]}{|w_c|}$$

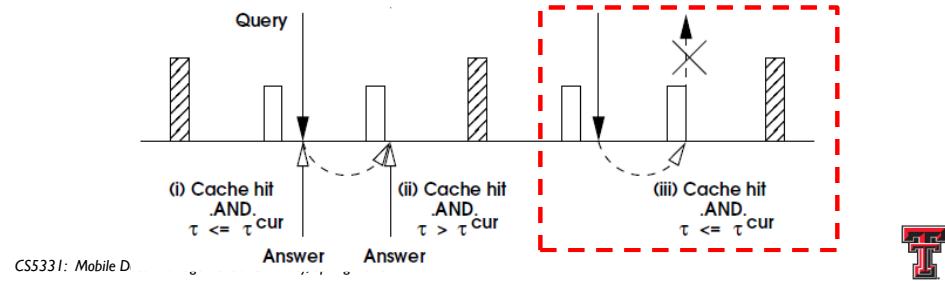


18

ConSens: Lazy Request

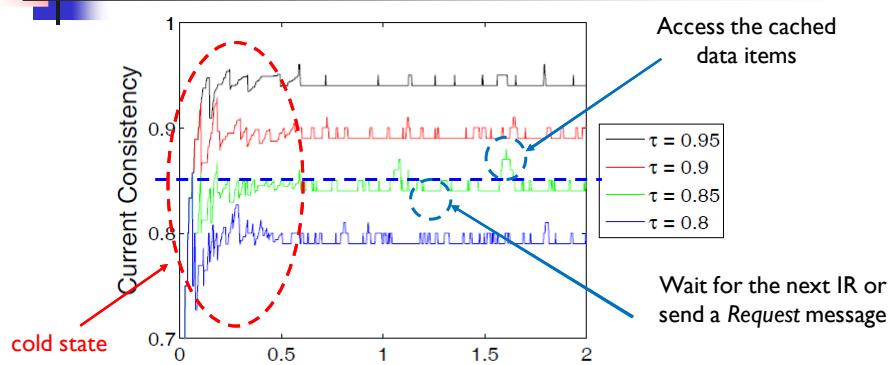
- Postpone sending an uplink Request message to the server, as far as
 - Consistency Condition ($\tau \leq \tau^{cur}$) is satisfied
 - Use an invalid cached data item for answering a query only one time
 - Update Invalid[u] and recalculate τ^{cur}

$$\tau^{cur} = 1 - \frac{\sum_{u=0}^{|w_c|} \text{Invalid}[u]}{|w_c|}$$



19

ConSens: Lazy Request



- Current Consistency (τ^{cur}) is fluctuating around the Target Consistency (τ)

CS5331: Mobile Data Management and Privacy, Spring 2023



20