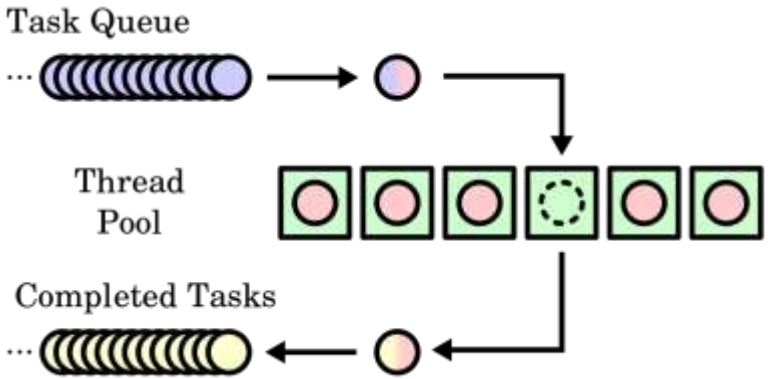
Caching, Pooling

Úvod



https://www.amd.com/en/processors/ryzen
By I, Cburnett, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2233464

Resource

- Resource
  - může být:
    - **reusable** x non-reusable
  - obrázek, záznam v DB, data
  - paměť
  - síťové spojení
- Resource User
- Resource Provider

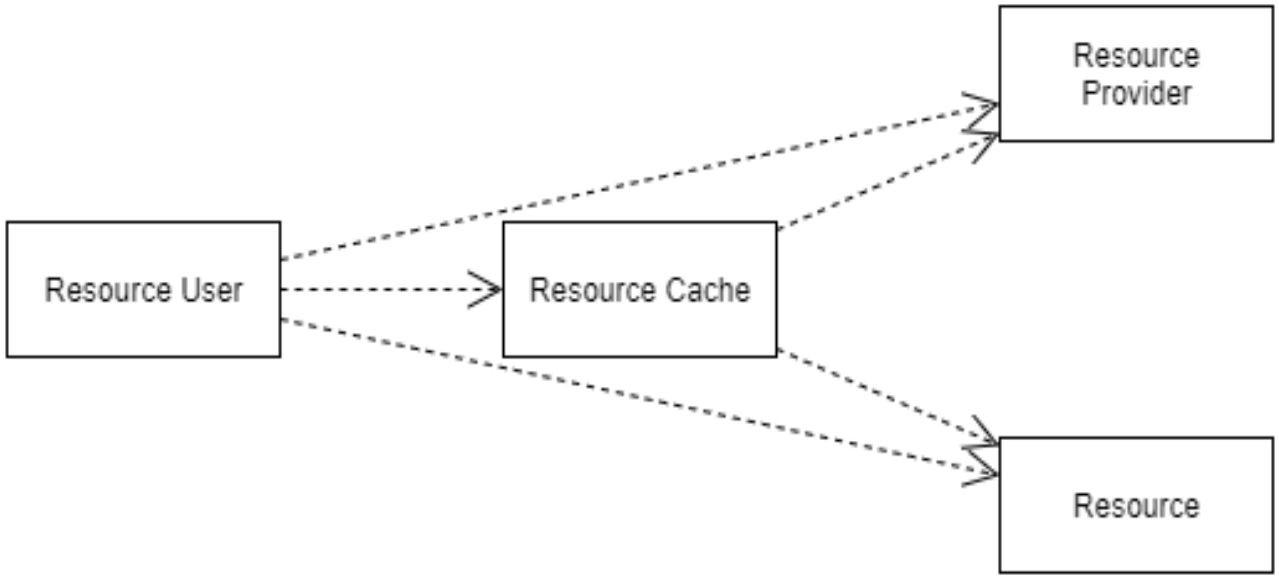
Caching

- Kontext
  - Opakovaný přístup k nějaké množině prostředků
  - Potřebujeme optimalizovat “výkon”
- Problém
  - **Režie** opakované akvizice, inicializace a uvolňování stejných prostředků
- Řešení
  - Buffer s rychlejším přístupem = **cache**
  - Při následujícím přístupu načteme z cache místo resource providera
  - (Identifikace, Eviction)

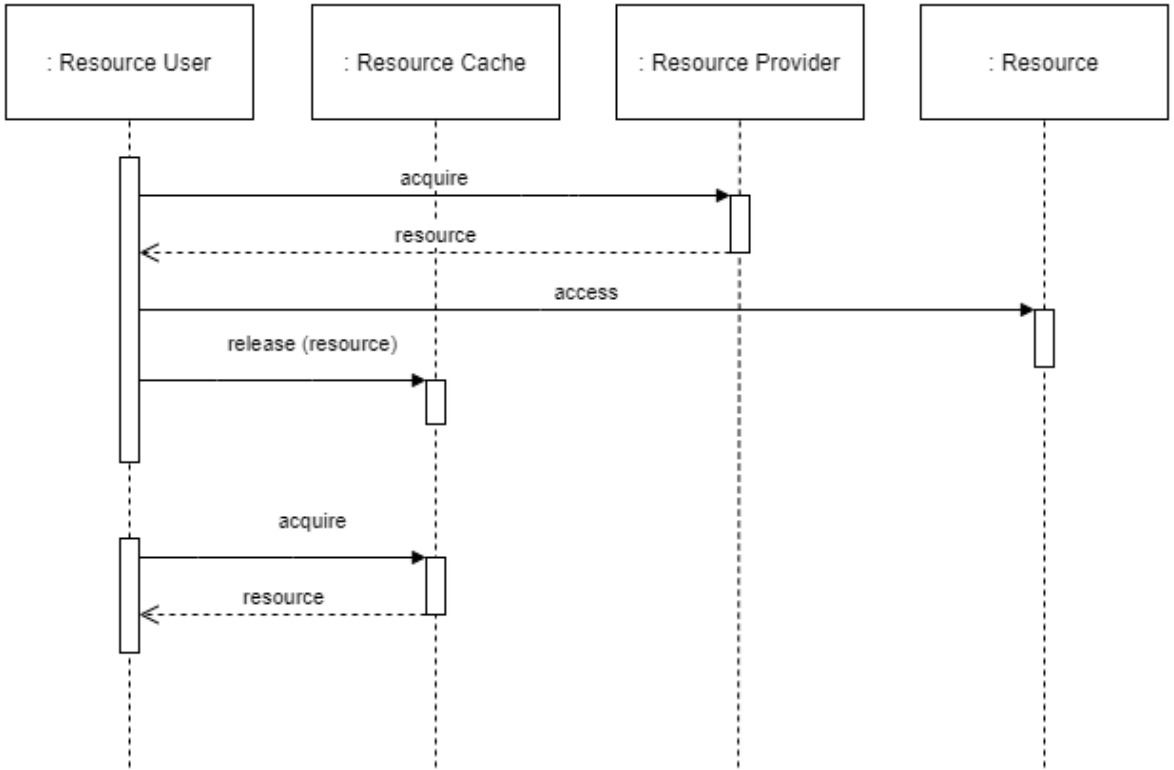
Caching - Příklad

- Píšeme aplikaci, která ukazuje stav síťových zařízení
- Periodicky se ptáme na jejich stav
- Nechceme neustále navazovat a ukončovat TCP spojení
- Místo uzavření je předáme cache
- Zajímá nás identita navázaných spojení!

Caching



Caching


: Resource User
: Resource Cache
: Resource Provider
: Resource
acquire
resource
access
release (resource)
acquire
resource

Caching - Recept Implementace podle POSA

1. Výběr prostředků, které stojí za to kešovat
2. Jak bude uživatel interagovat s cache?
3. Implementace cache
4. (Integrace cache)
    - Cache Proxy + Lazy Acquisition (POSA3)
    - Interceptor (POSA2)
5. Strategie pro vyhazování prvků z cache
    - Evictor(POSA3)
6. Zajištění konzistence
    - různé přístupy, např.:
        - cache-aside
        - write-through

Caching - Varianty

- Transparent Cache
  - Lazy Acquisition
- Read-Ahead Cache
  - Partial Acquisition
- Cached Pool
  - kombinace Caching a Pooling
  - dočasné zachování identity, poté recyklace
- Layered Cache
  - více vrstev
  - nějaká z vrstev může být sdílená

Caching - Pozitiva

- Výkon a škálovatelnost
  - latency x CPU x bandwidth
- Dostupnost
  - krátkodobý výpadek *resource providera*
- Stabilita
  - POSA: fragmentace paměti
    - moderní OS - virtuální x fyzická paměť
    - moderní jazyky - C#/Java CLR/JVM
      - přesouvání objektů
    - problém embedded, RTOS
  - garbage collection

Caching - Negativa

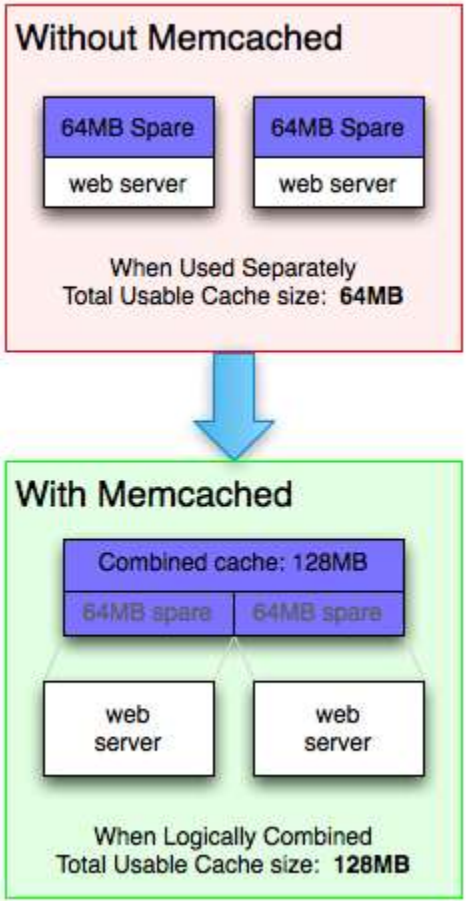
- Odolnost
  - pády SW, výpadky proudu
  - řešitelné synchronizací
- Vyšší spotřeba některých systémových zdrojů
- Implementace
  - synchronizace a konzistence, invalidace
  - tuning (kolik paměti)
- Zranitelnosti
  - DoS pomocí “cache poisoning”
  - DoS pomocí kolizí hashovací funkce
  - …

Caching - Využití

- Hardware cache
  - CPU - L1/L2/L3
  - HDD - buffer
  - SSD - buffer, menší SLC, větší TLC/QLC
- Webové prohlížeče
  - obrázky, styly, Javascript, …
- Sdílená HTTP cache
- Mapování souborů do paměti + stránkování
- Databáze
- Memoizace, Data
- Distribuované systémy

Caching - Využití 2

- Distribuovaná cache
  - Cache sdílená mezi více servery
    - Vlastní cache pro každou instanci je neefektivní
    - Oddělení životního cyklu serveru a cache
  - např.:
    - memcached
    - Redis



https://memcached.org/

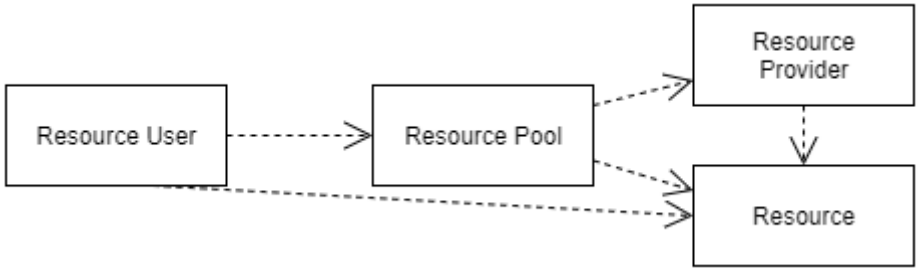
Pooling

- Kontext a problém:
  - Opakovaně vytváříme a ničíme prostředky stejného typu
  - Chtěli bychom je recyklovat
- Řešení:
  - Pool prostředků jednoho typu
  - Mohou být vytvořeny dopředu pomocí vzoru Eager Acquisition
  - Velikost může být dynamická

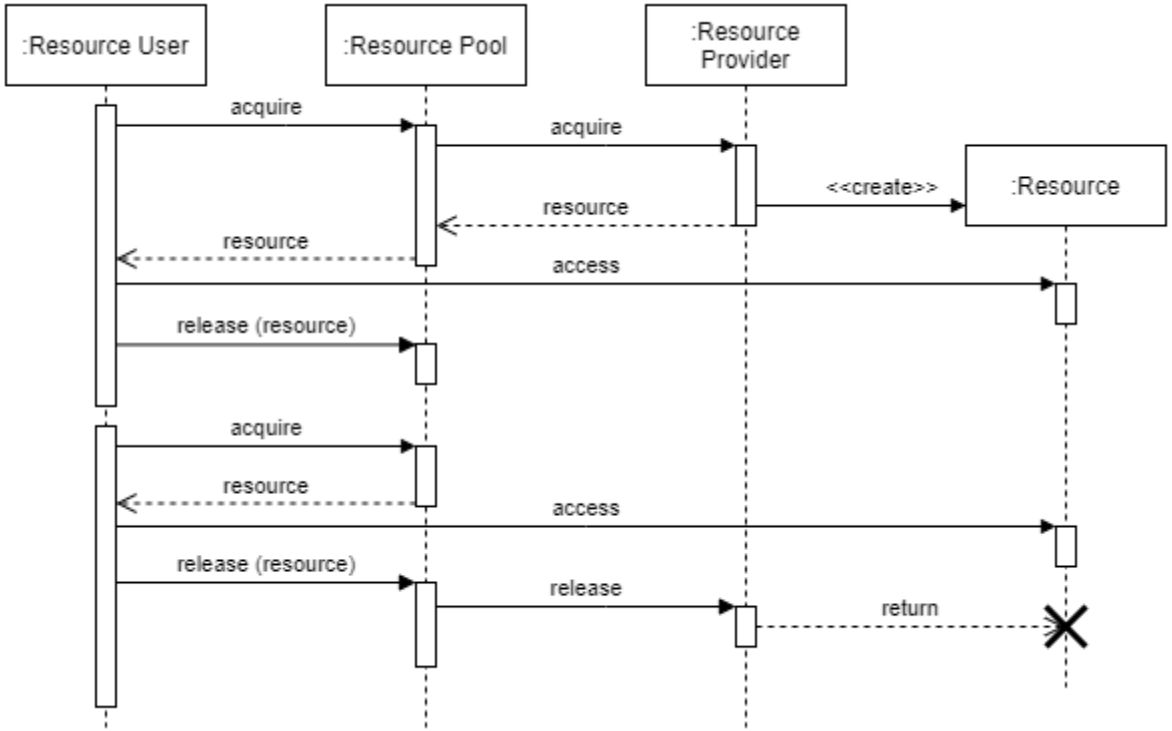
Pooling - Příklad

- Webový server s SQL databází
- Zpracováváme mnoho dotazů
- Vysoká režie otevření a zavření spojení s databází
- Budeme udržovat pool otevřených spojení

Pooling



Pooling


:Resource User
:Resource Pool
:Resource Provider
:Resource
acquire
acquire
<<create>>
resource
resource
access
release (resource)
acquire
resource
access
release (resource)
release
return

Pooling - Implementace

1. Výběr prostředků
2. Maximální velikost poolu
3. Počáteční velikost poolu
4. Rozhraní práce s prostředky
5. Rozhraní práce s poolem
6. Vyhazování prostředků z poolu
7. Jak se bude recyklovat?
    - například: přepsání paměti
8. Jak se bude řešit selhání?

Pooling - Varianty

- Mixed pool
  - různé druhy entit
- Sub-pools
  - rozdělení jednoho poolu na více menších poolů
  - různé sub-pooly k různým účelům
  - např. různá vlákna v ThreadPool

Pooling - Pozitiva

- Výkon
- Škálovatelnost
- Předvídatelnost
  - například omezení volání Garbage Collectoru

Pooling - Negativa

- Synchronizace
- Složitější použití
  - explicitní release
- Režie

Pooling - Příklady

- Thread pool
- JDBC Connection pool
- Memory pool
  - embedded, RTOS
- (Region-based memory, arena)
  - počítačová grafika
    - kniha PBR
- Object pooling
  - “reinkarnace instancí”
  - hry

Souvislosti s jinými návrhovými vzory

- Caching x Pooling
  - identita prostředků
  - konkrétní x nějaký
- Evictor (POSA3)
- Eager Acquisition (POSA3)
- Flyweight (GoF)
- Proxy (GoF), Cache Proxy
- Resource Lifecycle Manager (POSA3)
- Leasing (POSA3)
- Strategy
  - více algoritmů synchronizace stavu