Trabajo Práctico Final
Memcached

Sistemas Operativos I

Hedman Ulises

Pitinari Tomás

Universidad Nacional de Rosario
F.C.E.I.A.

1 Estructuras de datos internas

Las estructuras de datos internas se intentaron hacer genéricas para no tener que cambiar mucho en una posible reestructuracion de los datos.

1.1 NodeLL

```
struct _NodeLL {
        struct _NodeLL *backList, *nextList, *backLRU, *nextLRU;
        void *data;
};
```

Nodos de las estructuras **List** y **LRU**, guardan la direccion de memoria de sus vecinos y su propio dato, el cual va a ser un **NodeHT**.

1.2 List

```
struct _List {
        NodeLL rear, front;
};
```

Lista doblemente enlazada, con punteros al inicio y al final. Sus nodos son **NodeLL**.

1.3 LRU

```
struct _LRU {
        NodeLL rear, front;
        AllocationFunction custom_malloc;
        DestructiveFunction dest;
        InitDeallocateFunctionLRU preprocessing;
        EndDeallocateFunctionLRU postprocessing;
        OnAddElementLRU on_add_element;
        OnDeleteElementLRU on_delete_element;
        void *forwardRef;
};
```

Lista doblemente enlazada, con punteros al inicio y al final. Guarda funciones para manejar sus nodos y una referencia a su padre (estructura donde esta guardada la lista). Sus nodos son **NodeLL**.
Sirve principalmente para tener referencia del orden en el que se insertaron valores, para luego poder eliminarlos si hace falta memoria.

1.4 NodeHT

```
struct _NodeHT {
  void *key;
  unsigned keyLen;
  void *value;
  unsigned hashedKey;
};
```

Nodos que se guardan como dato de las listas de las casillas de la hash table. Se guarda la informacion de la clave y su tamaño, el valor y la clave ya hasheada.

1.5 HashTable

```
struct _HashTable {
  List *lists;
  pthread_mutex_t **lists_locks;
```

```
  LRU lru;
  pthread_mutex_t *lru_lock;
  unsigned size;
  atomic_int numElems;
};
```

Hash table, donde se guarda toda la informacion de los valores de la cache. **lists** es el arreglo de casillas, donde se encuentra una lista con los nodos para encontrarlos una vez se hashea la clave. **lru** es la LRU donde se guarda la informacion de los valores de la cache. **size** es la cantidad de casillas de la tabla. **numElems** es la cantidad de elementos actualmente en la tabla. Hay un lock por cada casilla y para la LRU.

2 Manejo de conexiones

El servidor se conecta con sus clientes por medio del protocolo TCP y genera hilos para escuchar los sockets. Una vez que hayan mensajes de los clientes, el servidor le envia a los hilos los file descriptors de dichos clientes asi estos pueden manejar los pedidos.
Los mensajes se leen progresivamente, asi si no se recive un mensaje completo, aun se puede procesar el pedido.

3 Política de desalojo

Los datos que se desalojan, son los primeros que se ingresaron, considerados como los mas viejos y necesarios de renovar. Para más eficiencia, se intentan desalojar hasta 10 datos a la vez, para no generar congestion en la LRU.