



Application Program Development

Segment : Thread pooling and JDBC Pooling

Presenter Name



Outcomes

- Understanding about thread pool.
- Understanding about JDBC pool.

Thread Pools – why?

- In the Java language, there are two ways to improve the execution efficiency of the program,
 - One is to use threads
 - The other is to use thread pools.
- In production environments, we usually use the latter.

Pooling Technology

- Pooling technology refers to preparing some resources in advance, and these pre-prepared resources can be reused when needed.
- There are two main advantages of pooling technology:
 - Preparation in advance
 - Reuse.
- Taking object creation in the Java language as an example, the following steps are required to create an object:
 - According to the parameters after the `new` identifier, look up the symbolic reference of the class in the constant pool.
 - If the symbol application is not found (the class is not loaded), load, parse, initialize the class, etc.
 - The virtual machine allocates memory in the heap for the object and initializes the allocated memory to 0. For the object header, the corresponding description structure is created (time-consuming operations: need to find free areas in the heap, modify the memory allocation status, etc.).
 - Call the initialization method of the object (time-consuming operation: complex logic verification of the user and other operations, such as IO, whether the numerical calculation conforms to the regulations, etc.).

Pooling Technology

- As can be seen from the above process, creating a class requires complex and time-consuming operations.
- Therefore, we should try to reuse the existing classes to ensure the efficient operation of the program.
- Of course, it would be great if these classes could be created in advance, and the implementation of these functions relies on pooling technology.

Application of Pooling Technology

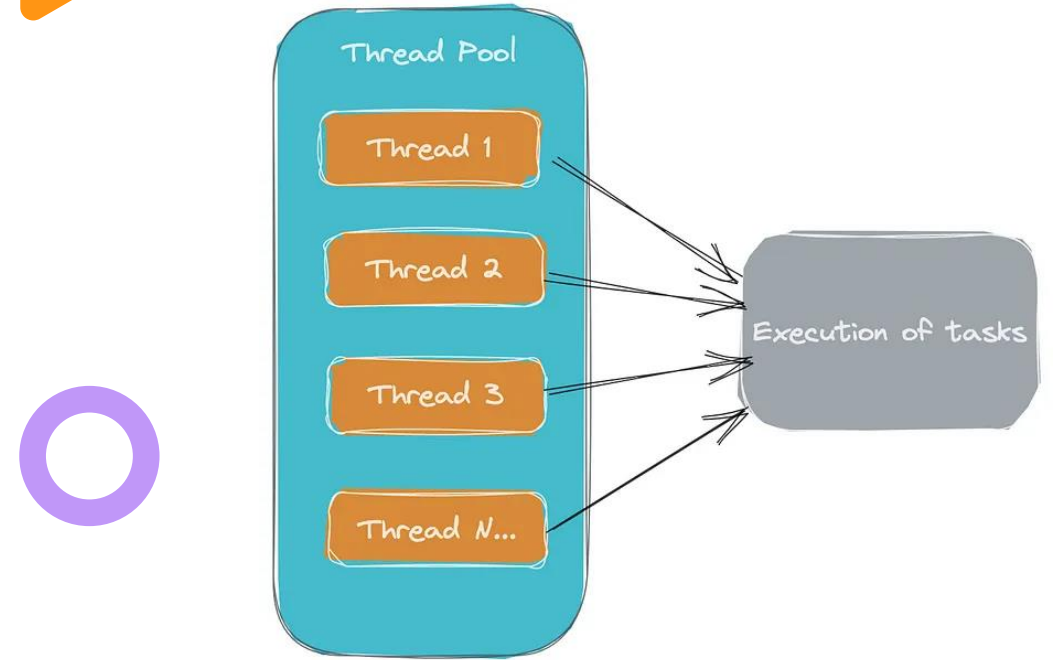
- Common applications of pooling technology include
 - **Thread pool**
 - **Memory pool**
 - **Database connection pool**
 - **HttpClient connection pool**, etc.

Thread Pool

- The principle of the thread pool is very simple, similar to the concept of buffers in the operating system.
 - A certain number of threads are started in the thread pool first, and these threads are all in the sleep state.
 - When the client has a new request, it will wake up a sleeping thread in the thread pool and let it process the client's request. After processing the request, the thread is in a sleep state again.
 - Thread pools can greatly improve program performance.
- For example, there is a bank network center with a large concentration of provincial data, and the number of concurrent client requests per second during the peak period exceeds 100.

Thread Pool

- If a new thread is created for each client request, the CPU time and memory consumption are staggering.
- If a thread pool with 200 threads is used, it will save a lot of system resources, so that more CPU time and memory are used to process actual business applications, instead of frequent thread creation and destruction.



Memory Pool

- How to better manage the use of application memory while increasing the frequency of memory use is a question worthy of every developer's pondering.
- Memory Pool (Memory Pool) provides a more feasible solution.
- In the process of creating a memory pool, enough memory is allocated in advance to form a preliminary memory pool.
- Then every time the user requests memory, a piece of free memory in the memory pool will be returned, and the mark of this memory will be set as used.
- When the memory is released after the memory is used, it is not the process of calling free or delete, but the process of putting the memory back into the memory pool, and the process of putting back the memory should set the flag to idle.
- Finally, when the application ends, the memory pool will be destroyed and every piece of memory in the memory pool will be released.

TCP and UDP

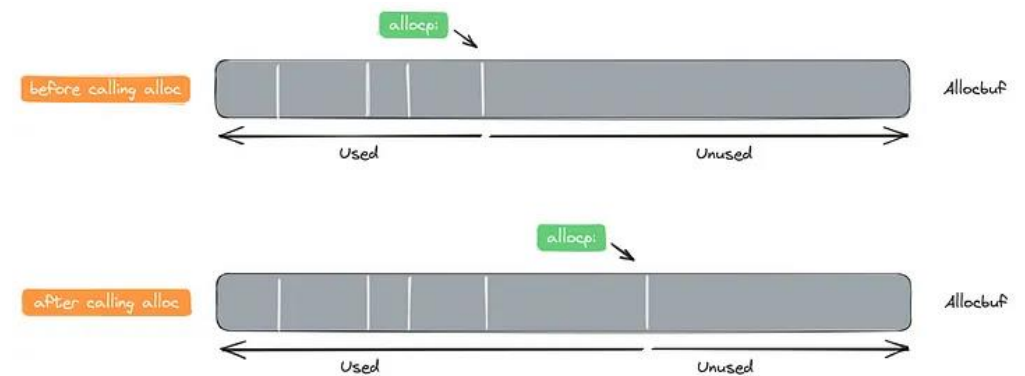
- TCP (*Transmission Control Protocol*): is a connection-based protocol that provides a reliable flow of data between two computers.
- UDP (*User Datagram Protocol*): is a protocol that sends independent packets of data, called datagrams, from one computer to another with no guarantees about arrival. UDP is not connection-based like TCP.

Advantages of memory Pool

- Reduce the generation of memory fragmentation.
- Increased memory usage frequency.

Disadvantage of memory pool

- It will cause a waste of memory because to use the memory pool needs to allocate a large piece of idle memory at the beginning, and these memories may not be used at all.

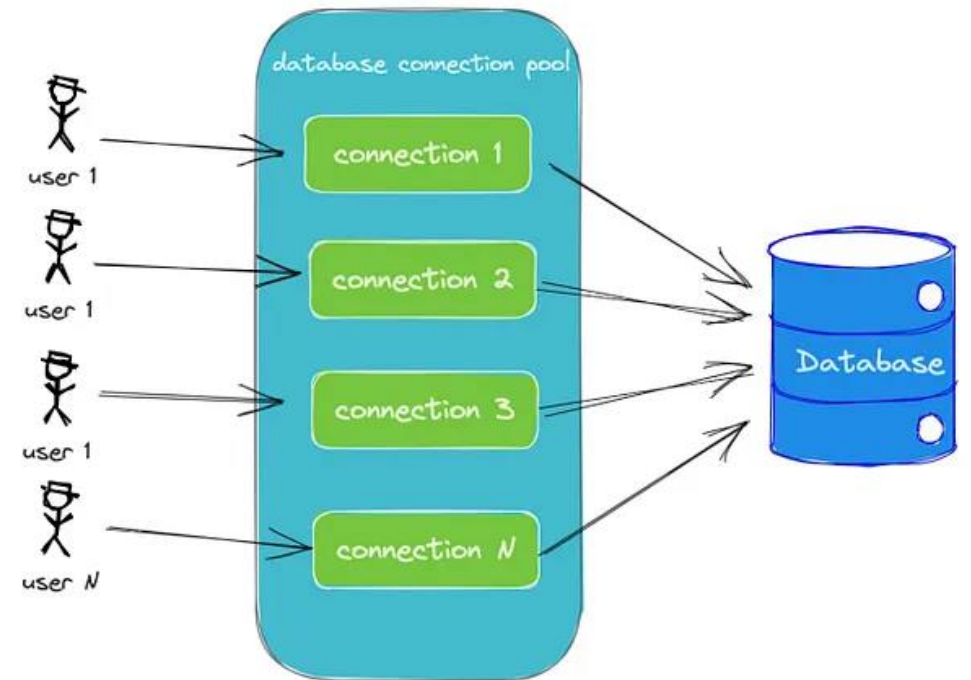


Database Connection Pool

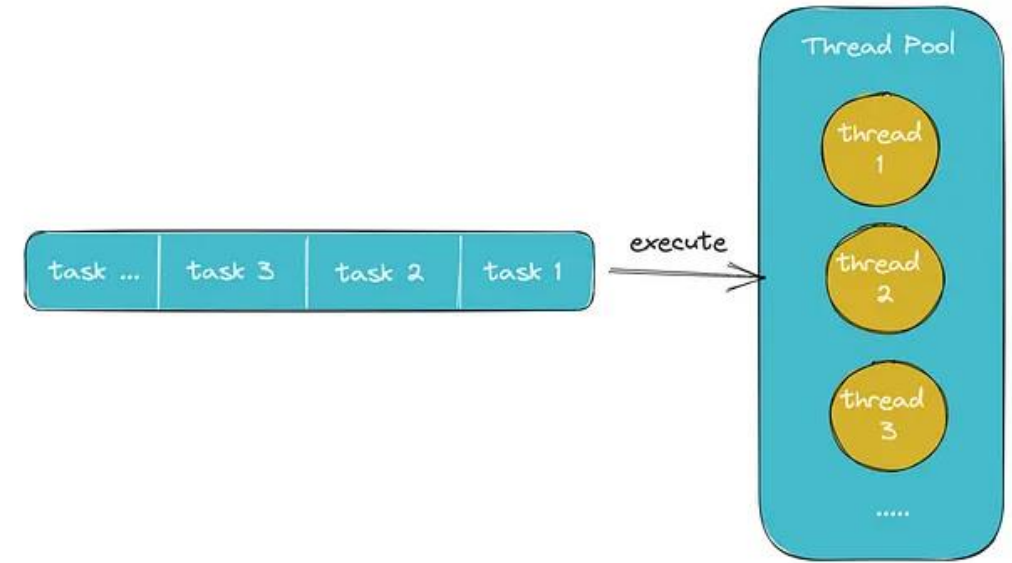
- The basic idea of the database connection pool is to store the database connection as an object in memory when the system is initialized.
- When the user needs to access the database, instead of establishing a new connection, an established idle connection is taken from the connection pool. object.
- After use, the user does not close the connection but puts the connection back into the connection pool for the next request to access, and the establishment and disconnection of these connections are managed by the connection pool itself.
- At the same time, you can also set the parameters of the connection pool to control the initial number of connections in the connection pool, the upper and lower limit of the connection, the maximum number of times each connection is used, and the maximum idle time.

Database Connection Pool

- The number and usage of connections can also be monitored through the management mechanism of the connection pool itself.



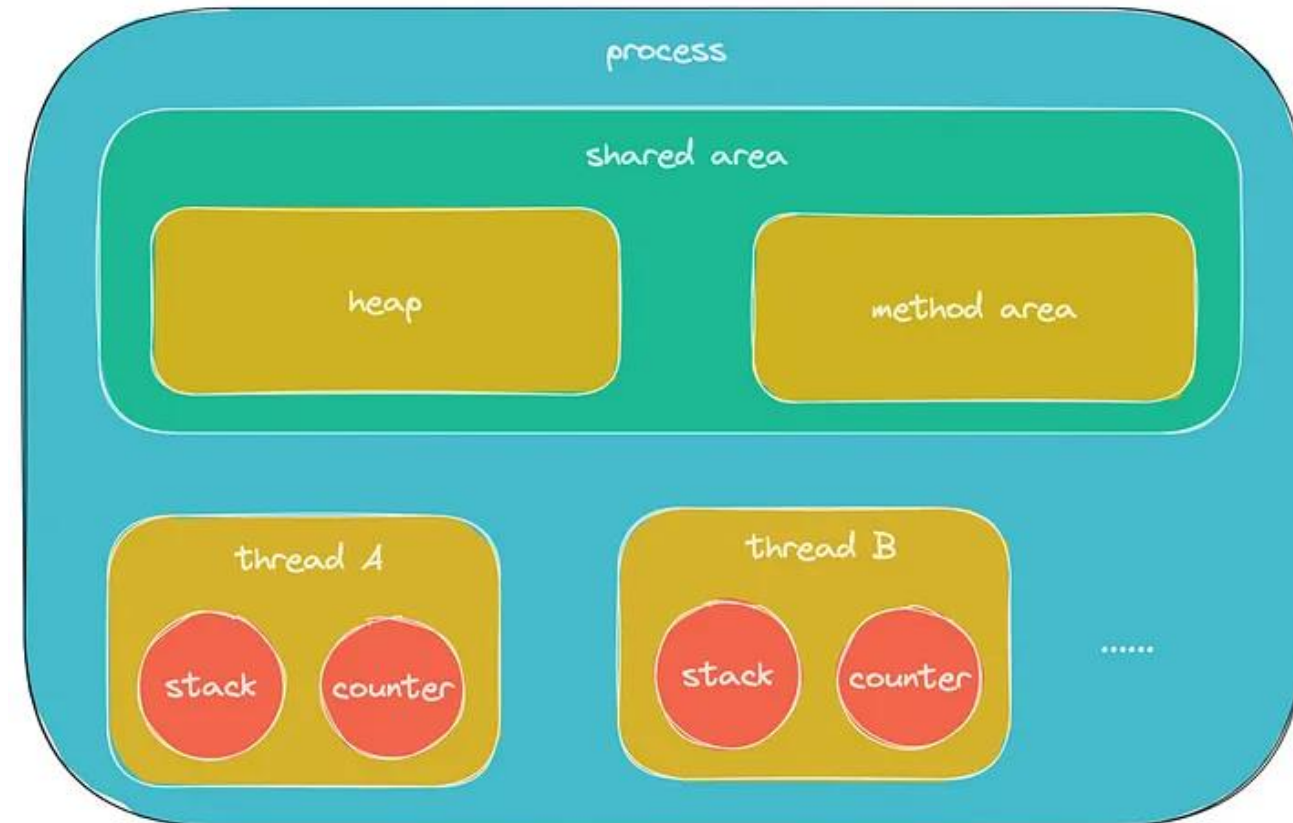
- A Thread pool is a pattern used by threads, which separates the concepts of threads and tasks, uses threads to perform tasks, and provides a unified implementation method for thread management and task management, avoiding the frequent creation and destruction of threads. performance overhead.
- Compared with threads, thread pools do not need to create and destroy threads frequently. Once a thread is created, it will remain in the thread pool by default.
- Wait until a task comes, and then use these **existing** threads to execute the **task**



Advantages of Thread Pools :

Reuse threads to reduce resource consumption.


- When a thread is created, it needs to open up the memory space of the virtual machine stack, native method stack, program counter, and other private threads.
- When destroyed, these private space resources are reclaimed
- After the thread pool creates a thread, it will be placed in the thread pool. Therefore, the first advantage of the thread pool compared to the thread is that it can reuse threads and reduce the consumption of system resources.





Advantages of Thread Pools :

Improve Responsiveness

- The thread pool reuses existing threads to perform tasks, and threads are newly created when there are tasks, so compared to threads, thread pools can respond to tasks and execute tasks faster.
- 



Advantages of Thread Pools :

Manage the number of threads and tasks

- Controlling the maximum number of concurrency: The thread pool can create a fixed number of threads, thus avoiding the problem of infinitely creating threads.
- 

Disadvantages of Thread Pools

- When too many threads are created, the system execution will slow down, because the number of CPU cores is certain and the number of tasks that can be processed at the same time is also certain.
- When there are too many threads, it will cause the problem of malicious thread contention and frequent thread switching, which will cause the program execution to slow down. Therefore, the appropriate number of threads is the key to the high-performance operation.
- Control the maximum number of tasks: If there are infinite tasks and insufficient memory, it will cause program execution errors.
- The thread pool can control the maximum number of tasks. When the number of tasks exceeds a certain number, a rejection strategy will be used to process the excess tasks, thus ensuring the healthy operation of the system.