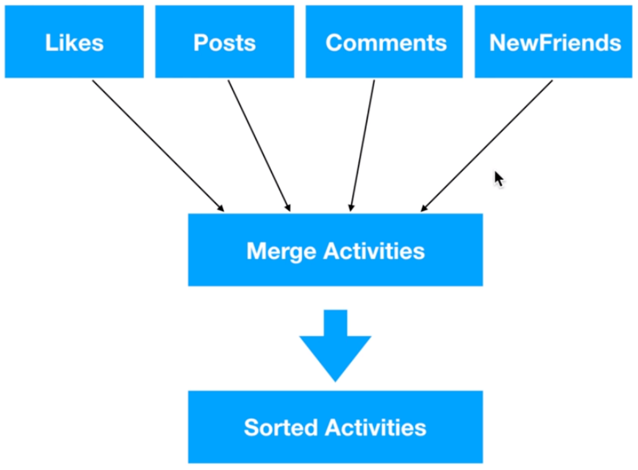
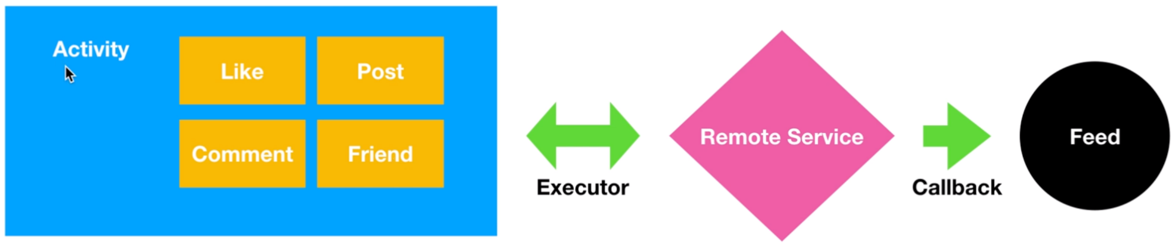
Problem Overview



- Building a user feed with list of activities in the order they happened.
- Likes, posts, comments, new friends are fetched through separate web APIs concurrently.
- When all the APIs result then the results are merged in a single sorted by date list.
- The result is returned to the caller through callback

Solution Design



Model Building Blocks

- Java utils concurrent package provide the tools to build up this model.
- ThreadPoolExecutor, Callable, Future

ThreadPoolExecutor

- It implement **ExecutorService.**
- It helps in using a fixed number of thread is running tasks and reusing the threads that has been done with the assigned task. This reduces the thread creation overhead.
- Create an **ThreadPoolExecutor** with a fixed number of threads in the pool using **Executors.newFixedThreadPool** method.
- Call **shutdown** method to stop all threads and release the resources.

Callable

- This is an **interface** used to return the result from the executing thread back to the invoking thread.
- **call** method is used to perform the task and then return the result or throw an exception.
- **Callable** task is run by the **ExecutorService** by calling its **submit** method.
- To run a **Runnable** task in **ExecutorService** we call its **execute** method.

Future

- A **Callable** task's result is returned by the **Future.**
- We get the result from the **Future** using its **get** method.
- The thread that calls the **get** method of the **Future** waits till the result is returned by the **Callable** after execution of its **call** method.

SOURCE CODE

https://blog.mindorks.com/java-android-multithreaded-programming-runnable-callable-future-executor