

Actors

4-18-2011

Opening Discussion

- Summer activities.
- Finishing the file example.

Challenges in Parallelism

- We have already seen that you need to have your programs multithreaded to take full advantage of modern processors.
- Unfortunately, the standard threading model can be very difficult to use. Race-conditions and deadlock can be difficult to avoid and are extremely challenging to debug.

Parallel Collections

- Starting with Scala 2.9, the collections library includes a parallel package.
- All applicable methods on the types in this package will run in parallel automatically. This means for loops will also run in parallel.
- You can convert any regular collection to a parallel collection with the `par` method.

Actors

- Another approach to parallelism is the actor model. This is the main approach to programming Erlang which is used a lot in telecom.
- Scala includes an actor package that supports this model.
- With actors, data isn't shared. You send messages instead of calling methods.
- Let's take a minute to look in the API.

Messages

- Actors respond to messages that are sent to them. The actions of actors are done in parallel.
- Use ! to send a message.
- It is ideal to use case classes for messages. That way you get pattern matching and immutable data.
- Messages go to an actors inbox. Sending messages does not block.

The act Method

- The main logic for an actor goes in the act method.
- Call receive to process a message.
- Put the receive inside a loop to do so repeatedly.
- Receive takes a partial function that handles the appropriate messages.
- Don't block.

Short Version

- There is a helper method in the Actor object that takes a function for the body of the act method.
- You should import `scala.actors.Actor`, then you can call `actor { body }`.

Playing with Actors

- Let's play a little with Actors given what we know so far.

Minute Essay

- Questions.