

Computación Paralela y Distribuida

ForkJoinPool

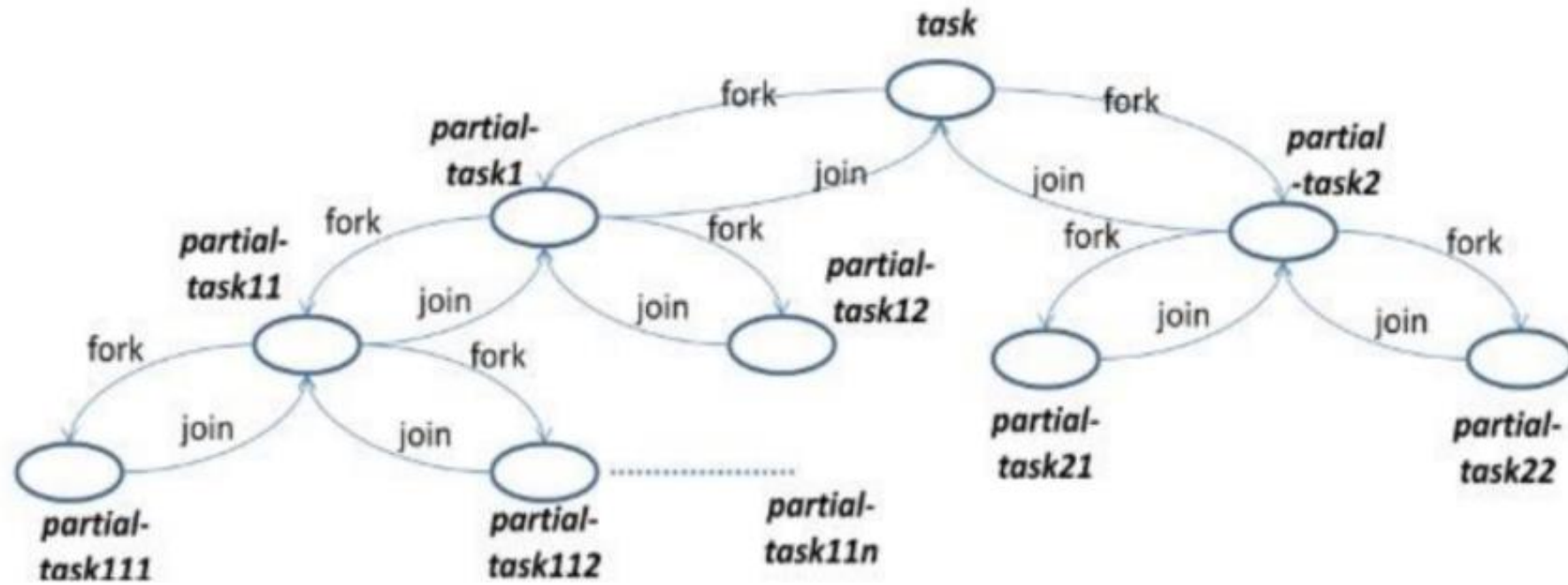
Diego Ordóñez Camacho, PhD.

ForkJoinPool

- Ejecutor capaz de crear nuevas tareas por sí mismo
 - Divide una tarea grande en otras más pequeñas (map)
 - Recolecta los resultados de las subtareas (reduce)
- Maximiza el uso de los hilos
 - Si la tarea que ejecuta un hilo queda en espera, el hilo busca otras tareas que ejecutar
 - Se las quita a otros hilos
- Con `invoke()` ejecuta la primera tarea, y devuelve el resultado global
- `java.util.concurrent`

RecursiveTask<T>

- Tarea que va a ser ejecutada
 - Se sobrescribe el método compute()
 - Este método devuelve un valor de tipo <T>
 - compute() es responsable de:
 - Si la tarea es manejable (suficientemente pequeña) hace el cálculo y devuelve el resultado
 - Si la tarea es muy grande la divide
 - Crea una o varias nuevas tareas y las arranca con fork()
 - Recupera los resultados parciales de las subtareas con join()
 - join() bloquea hasta que el resultado esté listo
 - Une el resultado de todas las subtareas y lo devuelve



How the Fork/Join framework uses divide-and-conquer to complete the task

