We are now in the multicore era. We have to write software that is easily parallelizable.

Driven by the need for elegant parallel operators, Java8 will introduce lambda expressions. The new Java8 Iterable<T> offers methods like forEach, map, filter, reduce that apply a lambda over the elements of the Iterable.

Parallelizing lambda-style operations is easy provided that iterations are independent. It involves just calling the parallel() method on the collection; the libraries will handle the rest.

Throwing away current software and starting to write parallel software from scratch is not an option.

Current code needs to be refactored; doing it manually is hard and error-prone.

We implement a NetBeans plugin that automatically refactors loops that iterate over collections to lambda expressions used in forEach, map, reduce, filter...

Design the tool such that the transformation preserves the original behavior of the program, i.e., is safe.

Develop the tool such that the transformation preserves the original behavior of the program, i.e., is safe.

Implement a NetBeans plugin that automatically refactors loops that iterate over collections to lambda expressions used in forEach, map, reduce, filter...

Design the tool such that the transformation preserves the original behavior of the program, i.e., is safe.

Transforming the code this way makes it really easy to make it run in parallel.

E.g., there are safety checks to make, and automating those checks could be an interesting extension.

Applying refactoring to forEach over five case studies showed it is widely applicable. When the refactoring wasn’t applied it was because the preconditions weren’t met.

Examples

```java
for (ReferenceType srcfa.getInstantiatedTypes()) {
  liveness.setLive(s, idx);
}
```

```java
forEach(c -> liveness.setLive(s, idx))
```

```java
for (ArgHandler argHandler : argHandlers.values()) {
  widest = handler.length();
}
```

```java
mapReduce(handler -> handler.length(), 'c', Integer::max);
```

```java
for (ReferenceType t : srcfa.getInstantiatedTypes())
  if (instanceof JDeclaredType)
    declared.add(t);
```

```java
cfa.getInstantiatedTypes().forEach(c -> liveness.setLive(s, idx))
```

```java
indexToTokenMap.values()
  .filter(map -> info->info.drawTokenBox)
  .forEach((info->drawToken((Graphics2D)g,Color.red,false));
```