

Exercices INF7440 : série #6 (bis)

1. Parallélisme itératif style «Map/Reduce» Le programme MPD qui suit présente l'interface d'un module qui définit des opérations `map` et `reduce`. Donnez une mise en oeuvre, la plus parallèle possible (parallélisme à granularité fine), de ces deux opérations.

```
global MapReduce

op type OpUnaire = ( int ) returns int;
op type OpBinaire = ( int, int ) returns int;

op map( cap OpUnaire f, int a[*], int n, res int r[*] );
# POSTCONDITION
# ALL( 1 <= i <= n :: r[i] = f( a[i] ) )

op reduce( cap OpBinaire o, int a[*], int n, res int r );
# NOTATION (infixe)
# Pour simplifier, on note "o( x, y )" par "x o y"
# PRECONDITION
# x o (y o z) = (x o y) o z = x o y o z
# POSTCONDITION
# r = a[1] o a[2] o a[3] o ... o a[n-1] o a[n]

body MapReduce
  import OpsAuxiliaires;

  proc map( f, a, n, r )
  {

}

  proc reduce( o, a, n, r )
  {
    assert( estPuissance(n, 2), "n pas une puissance de 2?!" );

}
end
```
