```
template<typename Range,typename Body> void parallel_reduce(const Range& range,const Body
```

```
template<typename Range,typename Body> auto parallel_reduce(const Range& range,const Body
-> typename std::enable_if<is_func_with_arg_type<Body, Range>::value, void >::type {… }
```

```
template<typename Range,typename Body> auto parallel_reduce(const Range& range,const Body
-> typename std::enable_if<!is_callable<Body>::value, void > ::type{
static_assert(Check_Assert<Body>::value,
``the 2nd parameter of parallel_reduce should be reference instead of pointer'');}
```