New Parallel Algorithms Contd Solutions

Copyright © James Raynard 2023

std::transform()

- Briefly describe the std::transform() function
  - std::transform() applies a callable object to each element in the input iterator range
  - It uses the results to populate the output iterator range
- Write a program to demonstrate its use

Copyright © James Raynard 2023

Binary overload of std::transform()

- Briefly describe the binary overload of std::transform()
  - The binary overload takes a second input iterator range
  - It applies a supplied function object to the corresponding elements from the two input ranges
  - It uses the results to populate the output range
- Write a program to demonstrate its use

Copyright © James Raynard 2023

Transform and Reduce Pattern

- Briefly describe the "transform and reduce" pattern
  - Divide the data into subsets
  - Start a thread for each subset
  - Each thread calls transform()
  - transform() performs some operation on the thread's subset
  - Call reduce() to combine each thread's results into the final answer

Copyright © James Raynard 2023

std:: inner_product

- Briefly describe the std::inner_product() function
  - std::inner_product multiplies the corresponding elements of two containers together
  - It returns the sum of these products
- Write a program to demonstrate its use

Copyright © James Raynard 2023

std::transform_reduce()

- Briefly describe the std::transform_reduce function
  - std::transform_reduce() is a re-implementation of std::inner_product() which supports execution policies
- Write a program to demonstrate its use

Copyright © James Raynard 2023

std::transform_reduce

- Why is transform_reduce particularly useful in parallel programming?
  - It can be used to implement the "map and reduce" pattern
- What are the advantages of combining std::transform() and std::reduce() into a single function?
  - The transform stages do not need to serialize their results
  - The reduce stage does not have to wait for all the transform stages to complete
  - The reduce stage can re-use threads from the transform stage

Copyright © James Raynard 2023