

Yaakoub Y El Khamra <yye00@cct.lsu.edu>

Date: 2005/11/14 05:17:40

Abstract

This thorn implement processor-local reduction operations.

1 Introduction

A reduction operation can be defined as an operation on arrays (tuples) of variables resulting in a single number. Typical reduction operations are sum, minimum/maximum value, and boolean operations. A typical application is, for example, finding the minimum value in an n -dimensional array.

This thorn provides processor-local reduction operations only. Global reduction operations can make use of these local reduction operations by providing the necessary inter-processor communication.

2 Numerical Implementation

The new local reduce thorn has several new features including index strides and offsets for array indexing and full complex number support. Pending request, weight support can be enabled (there are some issues that a mask is essentially a weight with 1 or 0 value).

Modifying or extending this thorn is quite a simple matter. The heart of all the reduction operations is the large iterator macro in `local_reductions.h`. This iterator supports n -dimensional arrays with offsets and strides. The iterator is used in all local reduction operators in this thorn. To add a reduction operator, or change an existing one, all that needs to be done is to change the actual reduction operation definition which is called from within the iterator to perform the reduction.

To use a custom local reduction operator from the new global reduction implementation, some values must be returned to the global reduction implementation, such as the type of MPI reduction operation that needs to be performed (MPI_SUM, MPI_MIN, MPI_MAX) and if the final result should include a division by the total number of points used in the reduction. These are set in the parameter table with keys: `mpi_operation` and `perform_division`.

3 Using This Thorn

Please refer to the `TestLocalReduce` thorn in the `CactusTest` arrangement.

4 Reduction Operations

4.1 Basic Reduction Operations

The following reduction operations are implemented. a_i are the values that are reduced, $i \in [1 \dots n]$.

count: The number of values

$$\text{count} := n$$

sum: The sum of the values

$$\text{sum} := \sum_i a_i$$

product: The product of the values

$$\text{product} := \prod_i a_i$$

sum2: The sum of the squares of the values

$$\text{sum2} := \sum_i a_i^2$$

sumabs: The sum of the absolute values

$$\text{sum2} := \sum_i |a_i|$$

sumabs2: The sum of the squares of the absolute values

$$\text{sumabs2} := \sum_i |a_i|^2$$

min: The minimum of the values

$$\text{min} := \min_i a_i$$

max: The maximum of the values

$$\text{max} := \max_i a_i$$

maxabs: The maximum of the absolute values

$$\text{maxabs} := \max_i |a_i|$$

Note that the above definitions are for both real and complex values. For $n = 0$, the result of the reduction operation is 0, except for product, which is 1, min, which is $+\infty$, and max, which is $-\infty$. We define the minimum of complex values by

$$\min(a + ib, x + iy) := \min(a, x) + i \min(b, y)$$

and define the maximum equivalently.

4.2 High-level Reduction Operations

The following high-level reduction operations are also implemented. They can be defined in terms of the basic reduction operations above.

average: The algebraic mean of the values

$$\text{average} := \text{sum}/\text{count}$$

norm1: The L_1 norm, i.e., the sum of the absolute values

$$\text{norm1} := \text{sumabs}/\text{count}$$

norm2: The L_2 norm, i.e., the Pythagorean norm

$$\text{norm2} := \sqrt{\text{sumabs2}/\text{count}}$$

norm_inf: The L_∞ norm

$$\text{norm_inf} := \text{maxabs}$$

4.3 Weighted Reduction Operations

It is often convenient to assign a weight w_i to each value a_i . In this case, the basic reduction operations are redefined as follows.

count: The number of values

$$\text{count} := \sum_i w_i$$

sum: The sum of the values

$$\text{sum} := \sum_i w_i a_i$$

product: The product of the values

$$\text{product} := \exp \sum_i w_i \log a_i$$

sum2: The sum of the squares of the values

$$\text{sum2} := \sum_i w_i a_i^2$$

sumabs: The sum of the absolute values

$$\text{sum2} := \sum_i w_i |a_i|$$

sumabs2: The sum of the squares of the absolute values

$$\text{sumabs2} := \sum_i w_i |a_i|^2$$

min: The minimum of the values

$$\min := \min_{i \mid w_i \neq 0} a_i$$

max: The maximum of the values

$$\max := \max_{i \mid w_i \neq 0} a_i$$

maxabs: The maximum of the absolute values

$$\maxabs := \max_{i \mid w_i \neq 0} |a_i|$$

The notation $\min_{i \mid w_i \neq 0} a_i$ means: “The minimum of a_i where i runs over all values where $w_i \neq 0$ ”. The definition of the high-level reduction operations does not change when weights are present.