**Table of Contents**

[1. About this document 2](#_Toc440893178)

[2. Concepts 2](#_Toc440893179)

[2.1. LessThan Comparable 2](#_Toc440893180)

[2.2. Strict Weak Ordering 2](#_Toc440893181)

[3. API Specifications 2](#_Toc440893182)

[3.1. HykSort 3](#_Toc440893183)

[3.1.1. For Normal Primitive Input Type 3](#_Toc440893184)

[3.1.2. For User-defined Input Type 6](#_Toc440893185)

[3.1.3. For Key-value Input Type 12](#_Toc440893186)

[3.1.3.1. With single value vector 12](#_Toc440893187)

[3.1.3.2. With multiple value vectors 16](#_Toc440893188)

[3.2. HyperQuickSort 92](#_Toc440893189)

[3.2.1. For Normal Input Type 92](#_Toc440893190)

[3.2.2. For Key-value Input Type 96](#_Toc440893191)

[3.2.2.1. With single value vector 96](#_Toc440893192)

[3.2.2.2. With multiple value vectors 100](#_Toc440893193)

[3.3. SampleSort 170](#_Toc440893194)

[3.3.1. For Normal Input Type 170](#_Toc440893195)

[3.3.2. For Key-value Input Type 174](#_Toc440893196)

[3.3.2.1. With single value vector 174](#_Toc440893197)

[3.3.2.2. With multiple value vectors 177](#_Toc440893198)

# 1. About this document

This document describes API specifications of the gpusort library. The API contains three main sections relate to three sorting algorithms: HykSort, HyperQuickSort, and SampleSort.

# 2. Concepts

## 2.1. LessThan Comparable

A type is LessThan Comparable if it is ordered: it must be possible to compare two objects of that type using operator<, and operator< must be a partial ordering.

## 2.2. Strict Weak Ordering

A Binary Function is a kind of function object: an object that is called as if it were an ordinary C++ function. A Binary Function is called with two arguments.

A Binary Predicate is a Binary Function whose result represents the truth or falsehood of some condition. A Binary Predicate might, for example, be a function that takes two arguments and tests whether they are equal.

A Strict Weak Ordering is a Binary Predicate that compares two objects, returning true if the first precedes the second. This predicate must satisfy the standard mathematical definition of a strict weak ordering. Some examples of this model are: less<T> and greater<T>. If comp is an object of class less<T> and x and y are objects of class T, then comp(x, y) returns true if x < y and false otherwise. Similarly, if comp is an object of class greater<T> and x and y are objects of class T, then comp(x, y) returns true if x > y and false otherwise.

# 3. API Specifications

The gpusort library’s API includes the following three sub-groups: API for using HykSort algorithm, API for using HyperQuickSort algorithm, and API for using SampleSort algorithm. Details of those sub-groups and their usage are described below.

## 3.1. HykSort

### 3.1.1. For Normal Primitive Input Type

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to sort the elements in a vector pointed by \_keys into ascending order, meaning that if i and j are any two valid elements in the vector such that i precedes j, then j is not less than i. This version compares objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the data vector.

Template Parameters:

Vector: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with a vector of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:



##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to sort the elements in a vector pointed by \_keys into an order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the data vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

Vector: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with a vector of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T>.

Example:



### 3.1.2. For User-defined Input Type

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to sort the elements of User-defined type in a vector pointed by \_list into ascending order, meaning that if i and j are any two valid elements in the vector such that i precedes j, then j is not less than i. This version compares objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

key\_func: A function pointer to get a key value from an object of the User-defined data type.

\_list: Address of the data vector.

Template Parameters:

KeyType: Type of keys.

KeyFunc: A function pointer type which its’ instance is used to get key value from an object of the User-defined data type .

Vector: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. Vector’s value\_type is a model of LessThan Comparable, and the ordering relation on Vector’s value\_type is a strict weak ordering.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

KeyType must be one of supported primitive types which are listed below:

int, unsigned int, long, unsigned long, float, double.

Example:





##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to sort the elements of User-defined type in a vector pointed by \_list into an order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

key\_func: A function pointer to get a key value from an object of the User-defined data type.

\_list: Address of the data vector.

Template Parameters:

KeyType: Type of keys.

StrictWeakOrdering: is a model of Strict Weak Ordering.

KeyFunc: A function pointer type which its’ instance is used to get key value from an object of the User-defined data type .

Vector: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. Vector’s value\_type is a model of LessThan Comparable, and the ordering relation on Vector’s value\_type is a strict weak ordering.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

KeyType must be one of supported primitive types which are listed below:

int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is Vector::value\_type.

Example:





### 3.1.3. For Key-value Input Type

#### 3.1.3.1. With single value vector

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort. That is, it sorts the elements in a vector pointed by \_keys and in a vector pointed by \_values into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and p and q are elements in the value vector corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values Address of the value vector.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:



##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort. That is, it sorts the elements in a vector pointed by \_keys and in a vector pointed by \_values into a key order in which comparing key objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values Address of the value vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



#### 3.1.3.2. With multiple value vectors

##### 3.1.3.2.1. Using C++11

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with variadic number of value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in variadic number of vectors pointed by corresponding pointers called \_values into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → +∞) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values Variadic pointers which each points to address of one value vector.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Vectors: Variadic templates of sequence container classes which each has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

To use this function, one must compile the gpusort library using C++11.

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:



##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with variadic number of value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in variadic number of vectors pointed by corresponding pointers called \_values into a key order in which comparing key objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values Variadic pointers which each points to address of one value vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Vectors: Variadic templates of sequence container classes which each has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

To use this function, one must compile the gpusort library using C++11.

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.1.3.2.2. Using an older version of C++

##### 3.1.3.2.2.1. With two value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in two value vectors pointed by \_values1 and \_values2 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 2) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:



##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in two value vectors pointed by \_values1 and \_values2 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.1.3.2.2.2. With three value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in three value vectors pointed by \_values1, \_values2, and \_values3 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 3) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:



##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in three value vectors pointed by \_values1, \_values2, and \_values3 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:





##### 3.1.3.2.2.3. With four value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in four value vectors pointed by \_values1, \_values2, \_values3, and \_values4 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 4) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:





##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in four value vectors pointed by \_values1, \_values2, \_values3, and \_values4 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.1.3.2.2.4. With five value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in five value vectors pointed by \_values1, \_values2, \_values3, \_values4 and \_values5 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 5) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:





##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in five value vectors pointed by \_values1, \_values2, \_values3, \_values4 and \_values5 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.1.3.2.2.5. With six value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in six value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5 and \_values6 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 6) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:





##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in six value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5 and \_values6 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.1.3.2.2.6. With seven value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in seven value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6 and \_values7 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 7) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:





##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in seven value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6 and \_values7 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.1.3.2.2.7. With eight value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in eight value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7 and \_values8 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 8) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:





##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in eight value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7 and \_values8 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.1.3.2.2.8. With nine value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in nine value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8 and \_values9 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 9) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:





##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in nine value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8 and \_values9 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.1.3.2.2.9. With ten value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in ten value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8, \_values9 and \_values10 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 10) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

\_values10 Address of value vector 10.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV10: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Example:





##### Customized Order

Syntax:



Description:

This function uses the HykSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in ten value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8, \_values9 and \_values10 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

k\_way: k-way value (a power of 2, ≤ number of MPI processes).

is\_using\_gpu: Using GPU devices to do local sort or not.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

\_values10 Address of value vector 10.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV10: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

ThrustException: An error occurred when executing a Thrust function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

This function must be used with vectors of primitive type. Supported primitive types are: int, unsigned int, long, unsigned long, float, double.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



## 3.2. HyperQuickSort

### 3.2.1. For Normal Input Type

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to sort the elements in a vector pointed by \_keys into ascending order, meaning that if i and j are any two valid elements in the vector such that i precedes j, then j is not less than i. This version compares objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the data vector.

Template Parameters:

Vector: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:



##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to sort the elements in a vector pointed by \_keys into an order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the data vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

Vector: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T>.

Example:



### 3.2.2. For Key-value Input Type

#### 3.2.2.1. With single value vector

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort. That is, it sorts the elements in a vector pointed by \_keys and in a vector pointed by \_values into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and p and q are elements in the value vector corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values Address of the value vector.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:



##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort. That is, it sorts the elements in a vector pointed by \_keys and in a vector pointed by \_values into a key order in which comparing key objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values Address of the value vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



#### 3.2.2.2. With multiple value vectors

##### 3.2.2.2.1. Using C++11

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with variadic number of value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in variadic number of vectors pointed by corresponding pointers called \_values into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → +∞) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values Variadic pointers which each points to address of one value vector.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Vectors: Variadic templates of sequence container classes which each has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

To use this function, one must compile the gpusort library using C++11.

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:



##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with variadic number of value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in variadic number of vectors pointed by corresponding pointers called \_values into a key order in which comparing key objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values Variadic pointers which each points to address of one value vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Vectors: Variadic templates of sequence container classes which each has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

To use this function, one must compile the gpusort library using C++11.

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.2.2.2.2. Using an older version of C++

##### 3.2.2.2.2.1. With two value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in two value vectors pointed by \_values1 and \_values2 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 2) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:



##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in two value vectors pointed by \_values1 and \_values2 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.2.2.2.2.2. With three value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in three value vectors pointed by \_values1, \_values2, and \_values3 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 3) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:



##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in three value vectors pointed by \_values1, \_values2, and \_values3 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:





##### 3.2.2.2.2.3. With four value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in four value vectors pointed by \_values1, \_values2, \_values3, and \_values4 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 4) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:





##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in four value vectors pointed by \_values1, \_values2, \_values3, and \_values4 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.2.2.2.2.4. With five value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in five value vectors pointed by \_values1, \_values2, \_values3, \_values4 and \_values5 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 5) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:





##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in five value vectors pointed by \_values1, \_values2, \_values3, \_values4 and \_values5 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.2.2.2.2.5. With six value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in six value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5 and \_values6 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 6) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:





##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in six value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5 and \_values6 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.2.2.2.2.6. With seven value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in seven value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6 and \_values7 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 7) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:





##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in seven value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6 and \_values7 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.2.2.2.2.7. With eight value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in eight value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7 and \_values8 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 8) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:





##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in eight value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7 and \_values8 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.2.2.2.2.8. With nine value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in nine value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8 and \_values9 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 9) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:





##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in nine value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8 and \_values9 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.2.2.2.2.9. With ten value vectors

##### Ascending Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in ten value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8, \_values9 and \_values10 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 10) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

\_values10 Address of value vector 10.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV10: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Example:





##### Customized Order

Syntax:



Description:

This function uses the HyperQuickSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in ten value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8, \_values9 and \_values10 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

\_values10 Address of value vector 10.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV10: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

The number of MPI processes must be a power of 2 (2n with n ≥ 0).

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



## 3.3. SampleSort

### 3.3.1. For Normal Input Type

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to sort the elements in a vector pointed by \_keys into ascending order, meaning that if i and j are any two valid elements in the vector such that i precedes j, then j is not less than i. This version compares objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the data vector.

Template Parameters:

Vector: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:



##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to sort the elements in a vector pointed by \_keys into an order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the data vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

Vector: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T>.

Example:



### 3.3.2. For Key-value Input Type

#### 3.3.2.1. With single value vector

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort. That is, it sorts the elements in a vector pointed by \_keys and in a vector pointed by \_values into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and p and q are elements in the value vector corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values Address of the value vector.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:



##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort. That is, it sorts the elements in a vector pointed by \_keys and in a vector pointed by \_values into a key order in which comparing key objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values Address of the value vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



#### 3.3.2.2. With multiple value vectors

##### 3.3.2.2.1. Using C++11

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with variadic number of value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in variadic number of vectors pointed by corresponding pointers called \_values into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → +∞) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values Variadic pointers which each points to address of one value vector.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Vectors: Variadic templates of sequence container classes which each has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

To use this function, one must compile the gpusort library using C++11.

Example:



##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with variadic number of value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in variadic number of vectors pointed by corresponding pointers called \_values into a key order in which comparing key objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values Variadic pointers which each points to address of one value vector.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Vectors: Variadic templates of sequence container classes which each has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

To use this function, one must compile the gpusort library using C++11.

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.3.2.2.2. Using an older version of C++

##### 3.3.2.2.2.1. With two value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in two value vectors pointed by \_values1 and \_values2 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 2) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:



##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in two value vectors pointed by \_values1 and \_values2 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.3.2.2.2.2. With three value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in three value vectors pointed by \_values1, \_values2, and \_values3 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 3) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:



##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in three value vectors pointed by \_values1, \_values2, and \_values3 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:





##### 3.3.2.2.2.3. With four value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in four value vectors pointed by \_values1, \_values2, \_values3, and \_values4 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 4) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:





##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in four value vectors pointed by \_values1, \_values2, \_values3, and \_values4 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.3.2.2.2.4. With five value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in five value vectors pointed by \_values1, \_values2, \_values3, \_values4 and \_values5 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 5) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:





##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in five value vectors pointed by \_values1, \_values2, \_values3, \_values4 and \_values5 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.3.2.2.2.5. With six value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in six value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5 and \_values6 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 6) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:





##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in six value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5 and \_values6 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.3.2.2.2.6. With seven value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in seven value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6 and \_values7 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 7) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:





##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in seven value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6 and \_values7 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.3.2.2.2.7. With eight value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in eight value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7 and \_values8 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 8) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:





##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in eight value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7 and \_values8 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.3.2.2.2.8. With nine value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in nine value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8 and \_values9 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 9) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:





##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in nine value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8 and \_values9 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:



##### 3.3.2.2.2.9. With ten value vectors

##### Ascending Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with two value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in ten value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8, \_values9 and \_values10 into ascending key order, meaning that if i and j are any two valid elements in the key vector such that i precedes j, and pk and qk are elements in the kth value vector (k = 1 → 10) corresponding to i and j respectively, then j is not less than i. This version compares key objects using operator<.

Parameters:

comm: MPI communicator (handle).

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

\_values10 Address of value vector 10.

Template Parameters:

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV10: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Example:





##### Customized Order

Syntax:



Description:

This function uses the SampleSort algorithm to perform a key-value sort with three value vectors on each process. That is, it sorts the elements in a vector pointed by \_keys and in ten value vectors pointed by \_values1, \_values2, \_values3, \_values4, \_values5, \_values6, \_values7, \_values8, \_values9 and \_values10 into a key order in which comparing objects is done by a function object comp.

Parameters:

comm: MPI communicator (handle).

comp: Comparison operator.

\_keys: Address of the key vector.

\_values1 Address of value vector 1.

\_values2 Address of value vector 2.

\_values3 Address of value vector 3.

\_values4 Address of value vector 4.

\_values5 Address of value vector 5.

\_values6 Address of value vector 6.

\_values7 Address of value vector 7.

\_values8 Address of value vector 8.

\_values9 Address of value vector 9.

\_values10 Address of value vector 10.

Template Parameters:

StrictWeakOrdering: is a model of Strict Weak Ordering.

VectorK: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector. VectorK’s value\_type is a model of LessThan Comparable, and the ordering relation on VectorK’s value\_type is a strict weak ordering.

VectorV1: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV2: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV3: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV4: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV5: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV6: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV7: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV8: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV9: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

VectorV10: A sequence container class which has full properties of std::vector in usage. Some examples of this type are: std::vector, thrust::host\_vector.

Exceptions:

MpiException: An error occurred when executing a MPI function used by the sorting algorithm.

Note:

Valid types to be used as StrictWeakOrdering are: std::less<T> or std::greater<T> where T is VectorK::value\_type.

Example:

