gather

Transfer distributed array or gpuArray to local workspace

Syntax

```
X = gather(A)
[X1,X2,...,Xn] = gather(A1,A2,...,,Xn)
X = gather(C,lab)
[X1,X2,...,Xn] = gather(C1,C2,...,Cn,lab)
```

Description

`X = gather(A)` can operate on the following array data:

- On a gpuArray: Transfers the elements of `A` from the GPU to the local workspace and assigns them to `X`.
- On a distributed array, outside an `spmd` statement: Gathers together the elements of `A` from the multiple workers to the local workspace and assigns them to `X`.
- On a codistributed array, inside an `spmd` statement or communicating job: Gathers together the elements of `A` and replicates them into `X` on every worker.

You can call `gather` on other data types, such as tall arrays (See `gather (tall)`). If the data type does not support gathering, then `gather` has no effect.

`X = gather(gpuArray(X))`, `X = gather(distributed(X))`, or `X = gather(codistributed(X))` return the original array `X`.

`[X1,X2,...,Xn] = gather(A1,A2,...,,Xn)` gathers multiple arrays `A1,A2,...,,An` into the corresponding outputs `X1,X2,...,Xn`. The number of input arguments and output arguments must match.

`X = gather(C,lab)` converts a codistributed array `C` to a variant array `X`, such that all of the elements are contained on worker `lab`, and `X` is a 0-by-0 empty double on all other workers.

`[X1,X2,...,Xn] = gather(C1,C2,...,Cn,lab)` gathers codistributed arrays `C1,C2,...,Cn` into corresponding outputs `X1,X2,...,Xn`, with all elements on worker `lab`. The number of input arguments and output arguments must match.

If the input argument to `gather` is not a distributed, a codistributed, or a gpuArray, the output is the same as the input.

Examples

collapse all

Gather gpuArrays

Gather the results of a GPU operation to the MATLAB® workspace.

```
G = gpuArray(rand(1024,1));
F = sqrt(G);   % Input and output are both gpuArray
W = gather(G); % Return array to workspace
whos
```

```
Name          Size        Bytes  Class

F          1024x1          108  gpuArray
G          1024x1          108  gpuArray
W          1024x1         8192  double
```

Gather Distributed Arrays

Gather all of the elements from a distributed array `D` onto the client.

```
n = 10;
D = distributed(magic(n)); % Distribute array to workers
M = gather(D)              % Return array to client
```

Gather Codistributed Arrays

Distribute a magic square across your workers, then gather the whole matrix onto every worker and then onto the client. This code results in the equivalent of `M = magic(n)` on all workers and the client.

```
n = 10;
spmd
  C = codistributed(magic(n));
  M = gather(C) % Gather all elements to all workers
end
S = gather(C)   % Gather elements to client
```

Gather all of the elements of `C` onto worker 1, for operations that cannot be performed across distributed arrays.

```
n = 10;
spmd
  C = codistributed(magic(n));
  out = gather(C,1);
  if labindex == 1
    % Characteristic sum for this magic square:
    characteristicSum = sum(1:n^2)/n;
    % Ensure that the diagonal sums are equal to the
    % characteristic sum:
    areDiagonalsEqual = isequal ...
      (trace(out),trace(flipud(out)),characteristicSum)
  end
end
```

```
Lab 1:
  areDiagonalsEqual =
    1
```

Tips

Note that `gather` assembles the codistributed or distributed array in the workspaces of all the workers on which it executes, or on the MATLAB client, respectively, but not both. If you are using `gather` within an `spmd` statement, the gathered array is accessible on the client via its corresponding `Composite` object; see Access Worker Variables with Composites. If you are running `gather` in a communicating job, you can return the gathered array to the client as an output argument from the task.

As the `gather` function requires communication between all the workers, you cannot gather data from all the workers onto a single worker by placing the function inside a conditional statement such as `if labindex == 1`.

See Also

`arrayfun` | `bsxfun` | `codistributed` | `distributed` | `gpuArray` | `pagefun`

Introduced in R2006b