# FLIP-14: crossGroup Operator

## Status

| | |
|---|---|
| **Discussion thread** | |
| **Vote thread** | |
| **JIRA** | *FLINK-1267* - *Add crossGroup operator* REOPENED |
| **Release** | |

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

## Motivation

A crossGroup operation is performed in most non-iterative Gelly algorithms including the similarity measures AdamicAdar and JaccardIndex as well as TriangleListing as a basis for clustering algorithms. Work is in-progress to add bipartite support to Gelly and each of the four projection methods will perform a crossGroup. A built-in operator will greatly reduce the complexity of these and future programs both in Gelly and for all Flink users.

crossGroup as a new operator will (as noted in FLINK-1267) work in lieu of reduceGroup (as in TriangleListing), which is memorybound and requires disabling object reuse or a copyable type, or a self-join, which duplicates the input and builds the full Cartesian product. The crossGroup operator can also optionally reduce the data skew caused by the quadratic expansion of distinct pairs (either n*n or n choose 2) with three operators as in JaccardIndex.

## Public Interfaces

GroupCrossHint enum with values OPTIMIZER_CHOOSES, UNIFORM_DISTRIBUTION, SKEWED_DISTRIBUTION.

GroupCrossFunction and FlatGroupCrossFunction similar to CrossFunction but bound to one input type parameter.

groupCross methods on SortedGrouping and UnsortedGrouping.

## Proposed Changes

Key considerations for discussion:

- whether to emit the full Cartesian product or only distinct pairs
- whether crossGroup is applicable to streaming

Each of the Gelly use cases only make use of distinct element pairs and so emitting the full Cartesian product require the UDF to ignore the unwanted half of the data by comparing the non-grouped fields. This is less efficient and requires that types implement Comparable. Given distinct pairs of element the full Cartesian product can be simulated by the UDF processing each pair both forwards and reversed, unioned with a map on the grouped DataSet; however, this eliminates any potential ordering on forwarded fields.

The crossGroup implementation will be similar to many existing Flink operators. For a uniform distribution, the CrossGroupDriver requires a spillable iterator which tracks two elements; also, in the case of emitting distinct elements the iterator can discard elements prior to the outer iterator. For a skewed distribution the operator will compile into three nodes. Similar to JaccardIndex, the first node will reduceGroup the Grouping and wrap each element in a Tuple3 with a group count and an increasing index. The second node will rebalance and flatMap each element and index into 1..count groups. The third node will implement a partial crossGroup where the initial group_size elements are emitted pairwise and with each following element.

## Compatibility, Deprecation, and Migration Plan

None required as this is a new feature.

## Test Plan

JaccardIndex (skewed distribution) and TriangleListing (uniform distribution) will be ported to use the new operator and the performance compared with the current ad hoc implementations.

## Rejected Alternatives

(none so far)