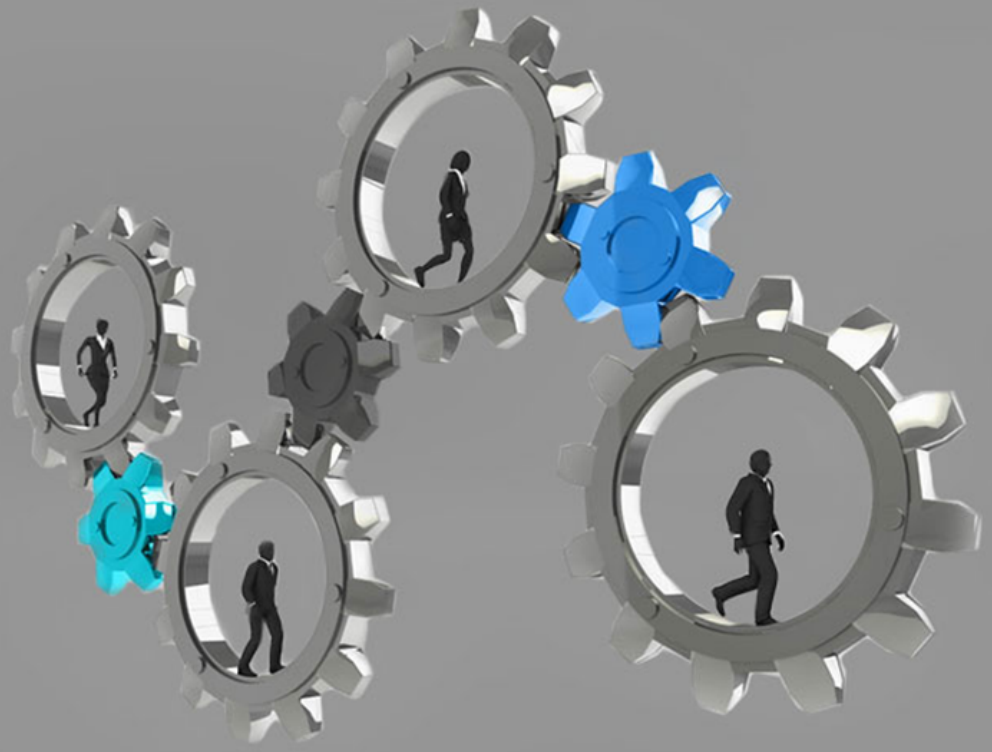
ENABLER OF CO-DESIGN



UCC Reduction Operation Types

Manjunath Gorentla Venkata, UCF Collectives WG,

July 29th, 2020

Reduction Operations : Support User-defined Types

```
typedef struct ucc_reduction_info {
    ucc_datatype_t      dt;
    ucc_reduction_op_t  op;
    size_t              count;
} ucc_reduction_info_t;

typedef enum {
    UCC_OP_MAX              = UCC_BIT(0),
    UCC_OP_MIN              = UCC_BIT(1),
    UCC_OP_SUM              = UCC_BIT(2),
    UCC_OP_PROD             = UCC_BIT(3),
    UCC_OP_AND              = UCC_BIT(4),
    UCC_OP_OR               = UCC_BIT(5),
    UCC_OP_XOR              = UCC_BIT(6),
    UCC_OP_LAND             = UCC_BIT(7),
    UCC_OP_LOR              = UCC_BIT(8),
    UCC_OP_LXOR             = UCC_BIT(9),
    UCC_OP_BAND             = UCC_BIT(10),
    UCC_OP_BOR              = UCC_BIT(11),
    UCC_OP_BXOR             = UCC_BIT(12),
    UCC_OP_MAXLOC           = UCC_BIT(13),
    UCC_OP_MINLOC           = UCC_BIT(14),
    UCC_OP_LAST_PREDEFINED  = UCC_BIT(15),
    UCC_OP_UNSUPPORTED      = UCC_BIT(16)
} ucc_reduction_op_t;
```

Reduction Operations : Support User-defined Types (Option1)

```
typedef struct ucc_reduction_info {
    ucc_datatype_t      dt;
    ucc_reduction_op_t  op;
    size_t              count;
    void                *user_defined_op;
} ucc_reduction_info_t;

typedef enum {
    UCC_OP_MAX              = UCC_BIT(0),
    UCC_OP_MIN              = UCC_BIT(1),
    UCC_OP_SUM              = UCC_BIT(2),
    UCC_OP_PROD             = UCC_BIT(3),
    UCC_OP_AND              = UCC_BIT(4),
    UCC_OP_OR               = UCC_BIT(5),
    UCC_OP_XOR              = UCC_BIT(6),
    UCC_OP_LAND             = UCC_BIT(7),
    UCC_OP_LOR              = UCC_BIT(8),
    UCC_OP_LXOR             = UCC_BIT(9),
    UCC_OP_BAND             = UCC_BIT(10),
    UCC_OP_BOR              = UCC_BIT(11),
    UCC_OP_BXOR             = UCC_BIT(12),
    UCC_OP_MAXLOC           = UCC_BIT(13),
    UCC_OP_MINLOC           = UCC_BIT(14),
    UCC_OP_LAST_PREDEFINED  = UCC_BIT(15),
    UCC_OP_USERDEFINED      = UCC_BIT(16),
    UCC_OP_UNSUPPORTED      = UCC_BIT(17)
} ucc_reduction_op_t;
```

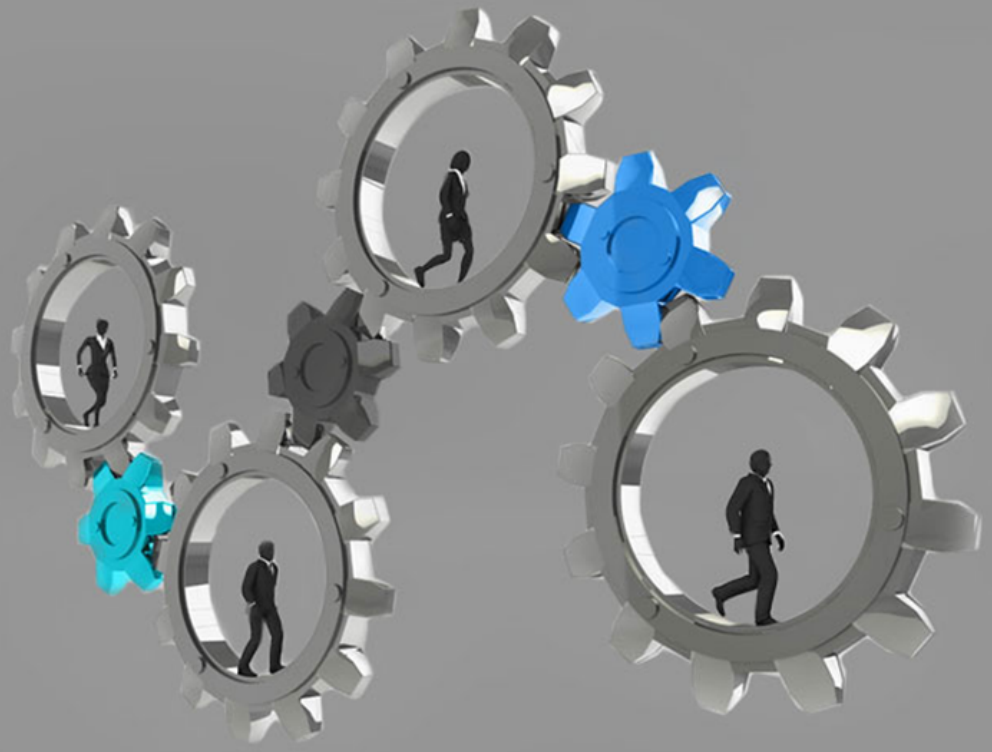
Reduction Operations : Support User-defined Types (Option2)

```
typedef struct ucc_reduction_info {
    ucc_datatype_t      dt;
    ucc_reduction_op_t  op;
    size_t              count;
    bool                user_defined_op;
} ucc_reduction_info_t;

typedef enum {
    UCC_OP_MAX              = UCC_BIT(0),
    UCC_OP_MIN              = UCC_BIT(1),
    UCC_OP_SUM              = UCC_BIT(2),
    UCC_OP_PROD             = UCC_BIT(3),
    UCC_OP_AND              = UCC_BIT(4),
    UCC_OP_OR               = UCC_BIT(5),
    UCC_OP_XOR              = UCC_BIT(6),
    UCC_OP_LAND             = UCC_BIT(7),
    UCC_OP_LOR              = UCC_BIT(8),
    UCC_OP_LXOR             = UCC_BIT(9),
    UCC_OP_BAND             = UCC_BIT(10),
    UCC_OP_BOR              = UCC_BIT(11),
    UCC_OP_BXOR             = UCC_BIT(12),
    UCC_OP_MAXLOC           = UCC_BIT(13),
    UCC_OP_MINLOC           = UCC_BIT(14),
    UCC_OP_LAST_PREDEFINED  = UCC_BIT(15),
    UCC_OP_UNSUPPORTED      = UCC_BIT(16)
} ucc_reduction_op_t;
```

ENABLER OF CO-DESIGN

UCF



Thank You

The UCF Consortium is a collaboration between industry, laboratories, and academia to create production grade communication frameworks and open standards for data centric and high-performance applications.