

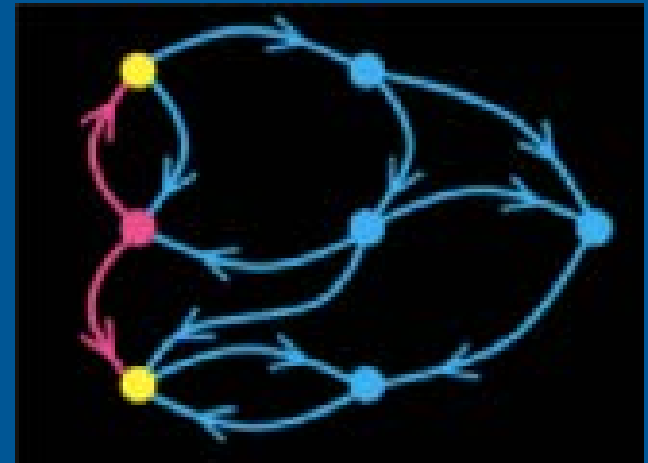
# GraphBLAS BoF @ HPEC17

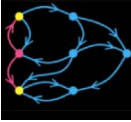
Co-chairs:

Aydin Buluç, LBNL

Marcin Zalewski, NIAC/PNNL

Scott McMillan, SEI/CMU





Copyright 2017 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

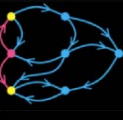
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

DM17-0627

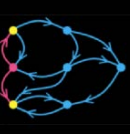
# Agenda



- The GraphBLAS Forum Overview
  - C API Specification “Report”
- Invited Speaker: Tim Davis, *SuiteSparse GraphBLAS*
- Short Topics
  - David Bader: *STINGER on GraphBLAS*
  - Andrew Lumsdaine: *A GraphBLAS Challenge: How I learned to stop worrying (about GraphBLAS) and love the C++ Standard Library*
- Open discussion



# GraphBLAS Forum Members

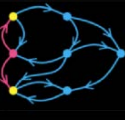


**Steering Committee:** David Bader, Aydın Buluç, John Gilbert, Jeremy Kepner, Tim Mattson, Henning Meyerhenke

## **Some other members (total count: 80+ as of August 2017):**

Peter (Zhang) Aaltonen, Hedayat Alghassi, Michael Anderson, Ariful Azad, Richard Barrett, Muthu Baskaran, Steven Dalton, Tim Davis, Joe Eaton, Franz Franchetti, Vijay Gadepally, Joseph Gonzales, Torsten Hoefler, Dylan Hutchison, Thejaka Kanewala, Manoj Kumar, Andrew Lumsdaine, Scott McMillan, Asit Mishra, José Moreira, Maxim Naumov, John Owens, Mostofa Ali Patwary, Fabrizio Petrini, Jason Riedy, Nadathur Satish, Narayanan Sunduram, Richard Veras, Michael Wolf, Carl Yang, Albert-Jan Yzelman, Marcin Zalewski, Xia Zhu

# GraphBLAS Forum Updates



Website: <http://graphblas.org>

- Hosted by UCSB
- Lists workshops and conferences
- Link to the latest C API Specification
- Lists teams developing implementations
- More information including the “The Math Document”

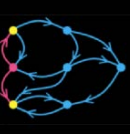
Mailing list: [Graphblas@lists.lbl.gov](mailto:Graphblas@lists.lbl.gov)

- Hosted by LBL (<mailto:abuluc@lbl.gov>)
- Join the Forum by joining the list

Monthly teleconference:

- Second Friday of every month, 12pm Eastern Time
- Send email to Jeremy Kepner to receive the calendar invite.

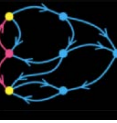
# GraphBLAS Forum Updates: C API



**C API Subcommittee:** Aydın Buluç, Tim Mattson, Scott McMillan, José Moreira, Carl Yang

## C API Specification Schedule:

- Version 1.0.2 released 10 August 2017
  - Minor update in the works (1.0.3) – send us typos, grammar issues, and bugs.
- Still a “provisional release” pending two conforming implementations.
- Conformance tests in planning (H1 2018?).



# C API Specification Overview

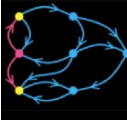
- Matrices and vectors as opaque types
  - Specify domain and size of dimensions (immutable)
  - Internal representation is hidden from user
  - They are not instantiated to be specifically sparse or dense.

```
GrB_Matrix graph;  
GrB_Matrix_new(&graph, GrB_BOOL, 10, 10);           // 10x10 matrix of Booleans
```

- Semirings used to describe operations (math document)
  - Some C functions can take monoids or arbitrary functions

```
GrB_Semiring arithmeticI32;           // <int32_t, float, float, +, *, 0, 1>  
GrB_Semiring_new(&arithmeticI32FF,  
                GrB_INT32, GrB_FLOAT, GrB_FLOAT,  
                GrB_PLUS_FLOAT, GrB_TIMES_I32FF, 0, 1);
```

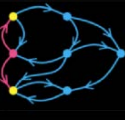
# GraphBLAS Base Operations



Operation	Math	Out	Inputs
mxm	$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle = \mathbf{C} \odot (\mathbf{A}^T \oplus \cdot \otimes \mathbf{B}^T)$	$\mathbf{C}$	$\neg, \mathbf{M}, \mathbf{z}, \odot, \mathbf{A}, \mathbf{T}, \oplus \cdot \otimes, \mathbf{B}, \mathbf{T}$
mxv (vxm)	$\mathbf{c}\langle \neg \mathbf{m}, \mathbf{z} \rangle = \mathbf{c} \odot (\mathbf{A}^T \oplus \cdot \otimes \mathbf{b})$	$\mathbf{c}$	$\neg, \mathbf{m}, \mathbf{z}, \odot, \mathbf{A}, \mathbf{T}, \oplus \cdot \otimes, \mathbf{b}$
eWiseMult	$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle = \mathbf{C} \odot (\mathbf{A}^T \otimes \mathbf{B}^T)$	$\mathbf{C}$	$\neg, \mathbf{M}, \mathbf{z}, \odot, \mathbf{A}, \mathbf{T}, \otimes, \mathbf{B}, \mathbf{T}$
eWiseAdd	$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle = \mathbf{C} \odot (\mathbf{A}^T \oplus \mathbf{B}^T)$	$\mathbf{C}$	$\neg, \mathbf{M}, \mathbf{z}, \odot, \mathbf{A}, \mathbf{T}, \oplus, \mathbf{B}, \mathbf{T}$
reduce (row)	$\mathbf{c}\langle \neg \mathbf{m}, \mathbf{z} \rangle = \mathbf{c} \odot [\oplus_j \mathbf{A}^T(:,j)]$	$\mathbf{c}$	$\neg, \mathbf{m}, \mathbf{z}, \odot, \mathbf{A}, \mathbf{T}, \oplus$
apply	$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle = \mathbf{C} \odot f(\mathbf{A}^T)$	$\mathbf{C}$	$\neg, \mathbf{M}, \mathbf{z}, \odot, \mathbf{A}, \mathbf{T}, f$
transpose	$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle = \mathbf{C} \odot \mathbf{A}^T$	$\mathbf{C}$	$\neg, \mathbf{M}, \mathbf{z}, \odot, \mathbf{A} (\mathbf{T})$
extract	$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle = \mathbf{C} \odot \mathbf{A}^T(\mathbf{i}, \mathbf{j})$	$\mathbf{C}$	$\neg, \mathbf{M}, \mathbf{z}, \odot, \mathbf{A}, \mathbf{T}, \mathbf{i}, \mathbf{j}$
assign	$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle (\mathbf{i}, \mathbf{j}) = \mathbf{C}(\mathbf{i}, \mathbf{j}) \odot \mathbf{A}^T$	$\mathbf{C}$	$\neg, \mathbf{M}, \mathbf{z}, \odot, \mathbf{A}, \mathbf{T}, \mathbf{i}, \mathbf{j}$
build (meth.)	$\mathbf{C} = \mathbb{S}^{\mathbf{m} \times \mathbf{n}}(\mathbf{i}, \mathbf{j}, \mathbf{v}, \odot)$	$\mathbf{C}$	$\odot, \mathbf{m}, \mathbf{n}, \mathbf{v}, \mathbf{i}, \mathbf{j}$
extractTuples (meth.)	$(\mathbf{i}, \mathbf{j}, \mathbf{v}) = \mathbf{A}$	$\mathbf{i}, \mathbf{j}, \mathbf{v}$	$\mathbf{A}$

Notation:  $\mathbf{i}, \mathbf{j}$  – index arrays,  $\mathbf{v}$  – scalar array,  $\mathbf{m}$  – 1D mask, **bold-lower** – vector (column), **bold-upper** – matrix,  $\mathbf{T}$  – transpose,  $\neg$  - structural complement,  $\mathbf{z}$  – replace  
 $\odot$  accumulate monoid/binary function,  $\oplus \cdot \otimes$  semiring,  
**blue** – optional parameters, **red** – optional modifiers (using Descriptors)

# GraphBLAS Signatures: mxm



$$\mathbf{C} \langle \neg \mathbf{M}, \mathbf{z} \rangle = \mathbf{C} \odot (\mathbf{A}^T \oplus \otimes \mathbf{B}^T)$$

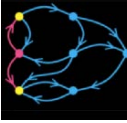
```
GrB_info GrB_mxm(GrB_Matrix          C,          // destination
                 const GrB_Matrix     Mask,
                 const GrB_BinaryFunction accum,
                 const GrB_Semiring    op,
                 const GrB_Matrix     A,
                 const GrB_Matrix     B,
                 const Descriptor      desc);
```

## Common Elements:

- GrB\_ “namespace”
- Destination (C) is first
- Mask matrix and accumulation function are next (if supported)
  - Pass GrB\_NULL if not needed.
- Descriptor is always last (**pass GrB\_NULL if not needed**)



# GraphBLAS Signatures: mxm



$$C \langle \neg M, z \rangle = C \odot (A^T \oplus \otimes B^T)$$

```
GrB_info GrB_mxm(GrB_Matrix C,  
                 const GrB_Matrix Mask,  
                 const GrB_BinaryFunction accum,  
                 const GrB_Semiring op,  
                 const GrB_Matrix A,  
                 const GrB_Matrix B  
                 const Descriptor desc);
```

## API Error Return Values:

- GrB\_NULL\_POINTER
- GrB\_INVALID\_VALUE
- GrB\_INVALID\_INDEX
- GrB\_DOMAIN\_MISMATCH
- GrB\_DIMENSION\_MISMATCH
- GrB\_OUTPUT\_NOT\_EMPTY
- GrB\_NO\_VALUE

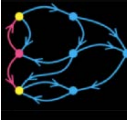
## Execution Error Return Values:

- GrB\_OUT\_OF\_MEMORY
- GrB\_INDEX\_OUT\_OF\_BOUNDS
- GrB\_PANIC

## Otherwise:

- GrB\_SUCCESS





# GraphBLAS Signatures: mxm

$$C\langle \neg M, z \rangle = C \odot (A^T \oplus \otimes B^T)$$

```

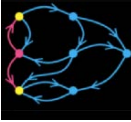
GrB_info GrB_mxm(GrB_Matrix      C,           // GrB_OUTP
                 const GrB_Matrix Mask,      // GrB_MASK
                 const GrB_BinaryFunction accum,
                 const GrB_Semiring op,
                 const GrB_Matrix A,         // GrB_ARG0
                 const GrB_Matrix B,         // GrB_ARG1
                 const Descriptor desc);

```

## Descriptors (modifiers to various arguments):

- GrB\_SCMP       $\neg$  – structural complement (MASK only).
- GrB\_TRAN       $T$  – transpose the input matrix arguments (ARG0, ARG1)
- GrB\_REPLACE  $z$  – replace all elements in the output argument (OUTP)





# Counting Triangles (once) with GraphBLAS

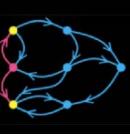
- Given:
  - Undirected graph  $G = \{V, E\}$
  - $L$ : boolean, lower-triangular portion of adjacency matrix
- **# triangles** =  $\|L \otimes (L \oplus \cdot \otimes L^T)\|_1$ 
  - Semiring can be Plus-AND or Plus-Times
  - Element-wise multiplication is equivalent to a mask operation

```
uint64_t triangle_count(GrB_Matrix L)    // L: NxN, lower-triangular, boolean
{
    GrB_Index N;
    GrB_Matrix_nrows(&N, L);
    GrB_Matrix C;
    GrB_Matrix_new(&C, GrB_UINT64, N, N);

    GrB_mxm(C, L, GrB_NULL, GrB_UInt64AddMul, L, L, GrB_TB); // C<L> = L * L^T

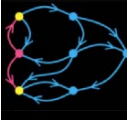
    uint64_t count;
    GrB_reduce(&count, GrB_NULL, GrB_UInt64Add, C, GrB_NULL); // 1-norm of C
    return count;
}
```





# Questions?





# Variants (assign, as an example)

- “Standard” variant (math document)

$$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle(\mathbf{i}, \mathbf{j}) \oplus = \mathbf{A}^T \qquad \mathbf{c}\langle \neg \mathbf{m}, \mathbf{z} \rangle(\mathbf{i}) \oplus = \mathbf{a}$$

- “Column and row” variants (on matrices only)

$$\text{row: } \mathbf{C}\langle \neg \mathbf{m}^T, \mathbf{z} \rangle(\mathbf{i}, \mathbf{j}) \oplus = \mathbf{a}^T \qquad \text{col: } \mathbf{C}\langle \neg \mathbf{m}, \mathbf{z} \rangle(\mathbf{i}, \mathbf{j}) \oplus = \mathbf{a}$$

- “Constant” variant

$$\mathbf{C}\langle \neg \mathbf{M}, \mathbf{z} \rangle(\mathbf{i}, \mathbf{j}) \oplus = s \qquad \mathbf{c}\langle \neg \mathbf{m}, \mathbf{z} \rangle(\mathbf{i}) \oplus = s$$

