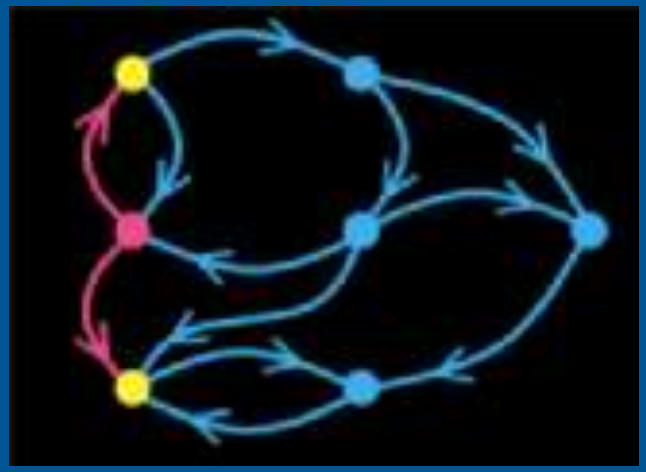
GraphBLAS BoF @ HPEC18

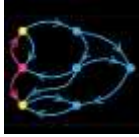


Co-chairs:

Tim Mattson, Intel

Marcin Zalewski, NIAC/PNNL

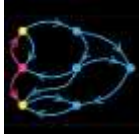
Scott McMillan, SEI/CMU



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.

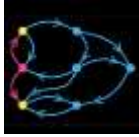
DM18-1095

Agenda



- The GraphBLAS Forum Overview
  - C API Specification “Report”
- Invited Speaker: Roi Lipman, *Redis Lab*
  - *“Sparse Matrices: a Missing Piece in Graph Databases”*
- Short Topics
  - Tim Davis: *SuiteSparse:GraphBLAS Update*
- Open discussion

GraphBLAS Forum Members

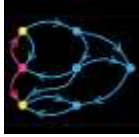


Steering Committee: David Bader, Aydın Buluç, John Gilbert, Jeremy Kepner, Tim Mattson, Henning Meyerhenke

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 Information



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”

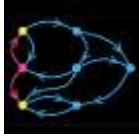
Mailing list: Graphblas@lists.lbl.gov

- 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

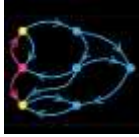


C API Subcommittee: Aydın Buluç, Tim Mattson, Scott McMillan, José Moreira, Carl Yang

C API Specification Schedule:

- Version 1.2.0 released May 2018
  - Removed “provisional” with two complete implementations.
- Acceptance tests still under construction

GraphBLAS Forum Updates: C API

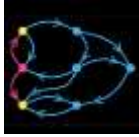


C API Subcommittee: Aydın Buluç, Tim Mattson, Scott McMillan, José Moreira, Carl Yang

C API Specification Schedule:

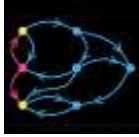
- Version 1.2.0 released May 2018
  - Removed “provisional” with two complete implementations.
- Acceptance tests still under construction
- Regular specification committee meetings still held
  - Answer questions regarding the specification language
  - Discuss proposals for changes and extensions
- No schedule for version 2.0

GraphBLAS Implementations



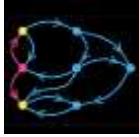
- SuiteSparse:GraphBLAS
  - Tim Davis, Texas A&M University (giving update tonight)
  - http://faculty.cse.tamu.edu/davis/suitesparse.html
- The IBM GraphBLAS C implementation
  - Jose Moreira, et al., IBM Research
  - https://github.com/IBM/ibmgraphblas
- C-wrapper around the GPU Gunrock library (in progress)
  - Carl Yang, UC Davis, http://adsabs.harvard.edu/abs/2017arXiv170101170W
- GraphBLAS Template Library v 2.0, C++ API
  - Scott McMillan, Andrew Lumsdaine, et al. (CMU/SEI, PNNL, UW)
  - https://github.com/cmu-sei/gbtl
- pyGB, python API (wrapper around GBTL)

Agenda



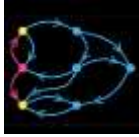
- The GraphBLAS Forum Overview
  - C API Specification “Report”
- **Invited Speaker: Roi Lipman, *Redis Lab***
  - ***“Sparse Matrices: a Missing Piece in Graph Databases”***
- Short Topics
  - Tim Davis: *SuiteSparse:GraphBLAS Update*
- Open discussion

Agenda



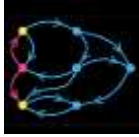
- The GraphBLAS Forum Overview
  - C API Specification “Report”
- Invited Speaker: Roi Lipman, *Redis Lab*
  - *“Sparse Matrices: a Missing Piece in Graph Databases”*
- **Short Topics**
  - **Tim Davis: *SuiteSparse:GraphBLAS Update***
- Open discussion

Agenda



- The GraphBLAS Forum Overview
  - C API Specification “Report”
- Invited Speaker: Roi Lipman, *Redis Lab*
  - *“Sparse Matrices: a Missing Piece in Graph Databases”*
- Short Topics
  - Tim Davis: *SuiteSparse:GraphBLAS Update*
- **Open discussion**

Open Discussion

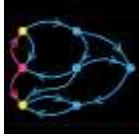


Possible future directions:

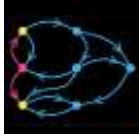
- Parallel GraphBLAS with a formal memory model for execution with multiple threads.
- The ability to enquire about the data format for GraphBLAS matrices to make it easier to interface to third party libraries.
- How does the API need to change to move into an MPI environment?

<insert your topic here>

Backups



C API Specification Overview



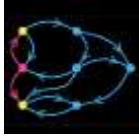
- Matrices and vectors as opaque types
  - Specify domain and size of dimensions (immutable)
  - Internal representation is hidden from user
  - They are not instantiated to be specifically sparse or dense.

```
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);
```

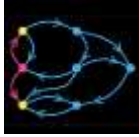
GraphBLAS Base Operations



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

Notation: **i**,**j** – index arrays, **v** – scalar array, **m** – 1D mask, **bold-lower** – vector (column), **M** – 2D mask, **bold-upper** – matrix, **T** – transpose, **¬** - structural complement, **z** – replace ⊙ accumulate monoid/binary function, ⊕.⊗ semiring, **blue** – optional parameters, **red** – optional modifiers (using Descriptors)

GraphBLAS Signatures: mxm



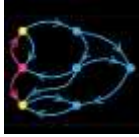
$$C\langle \neg M, z \rangle = C \odot (A^T \oplus.\otimes B^T)$$

```
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

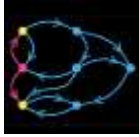
Execution Error Return Values:

- GrB_OUT_OF_MEMORY
- GrB_INDEX_OUT_OF_BOUNDS
- GrB_PANIC

Otherwise:

- GrB_SUCCESS

GraphBLAS Signatures: mxm



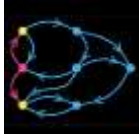
$$C\langle \neg M, z \rangle = C \odot (A^T \oplus.\otimes B^T)$$

```
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):

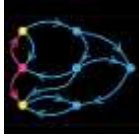
- GrB_SCMP ¬ – structural complement (MASK only).
- GrB_TRAN ^{T} – transpose the input matrix arguments (ARG0, ARG1)
- GrB_REPLACE z – replace all elements in the output argument (OUTP)

Counting Triangles (once) with GraphBLAS



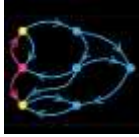
- Given:
  - Undirected graph G = {V, E}
  - L: boolean, lower-triangular portion of adjacency matrix
- **# triangles = $\|L \otimes (L \oplus.\otimes L^T)\|_1$**
  - Semiring can be Plus-AND or Plus-Times
  - Element-wise multiplication is equivalent to a mask operation

```
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)

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

- "Column and row" variants (on matrices only)

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

- "Constant" variant

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