Using Model Checking with Symbolic Execution to Verify Parallel Numerical Programs

Stephen F. Siegel[1]
Anastasia Mironova[2] George S. Avrunin[1] Lori A. Clarke[1]

[1]University of Massachusetts Amherst

[2]University of Utah

ISSTA 2006
Portland, Maine, July 17–20, 2006

Parallel numerical programs

$$(x_1, \ldots, x_n) \longrightarrow \boxed{P} \longrightarrow (y_1, \ldots, y_m)$$

Parallel numerical programs

$$(x_1, \ldots, x_n) \longrightarrow \boxed{\text{P}} \longrightarrow (y_1, \ldots, y_m)$$

- scientific computation
  - simulations of physical phenomena
  - climate modeling
  - molecular dynamics

Parallel numerical programs

$$(x_1, \dots, x_n) \longrightarrow \boxed{\mathsf{P}} \longrightarrow (y_1, \dots, y_m)$$

- scientific computation
  - simulations of physical phenomena
  - climate modeling
  - molecular dynamics
- matrix algorithms
  - solving systems of linear equations
  - matrix factorization
  - reducing to normal forms

Parallel numerical programs

$$(x_1, \ldots, x_n) \longrightarrow \boxed{\text{P}} \longrightarrow (y_1, \ldots, y_m)$$

- scientific computation
  - simulations of physical phenomena
  - climate modeling
  - molecular dynamics
- matrix algorithms
  - solving systems of linear equations
  - matrix factorization
  - reducing to normal forms

The Problem:

Parallel numerical programs are very difficult to get right.

Matrix multiplication: sequential version

```
double A[N][L], B[L][M], C[N][M];
            ⋮
int i,j,k;
for (i=0; i<N; i++)
  for (j=0; j<M; j++) {
    C[i][j] = 0.0;
    for (k=0; k<L; k++)
      C[i][j] += A[i][k]*B[k][j];
  }
```

Matrix multiplication: parallel version (master-slave)

```
int rank,nprocs,i,j,numsent,sender,row,anstype;
double buffer[L], ans[M];
MPI_Status status;
MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
if (rank==0) { /* I am the master */
  numsent=0;
  for (i=0; i<nprocs-1; i++) {
    for (j=0; j<L; j++)
      buffer[j] = A[i][j];
    MPI_Send(buffer, L, MPI_DOUBLE, i+1,
      i+1, MPI_COMM_WORLD);
    numsent++;
  }
  for (i=0; i<N; i++) {
    MPI_Recv(ans, M, MPI_DOUBLE, MPI_ANY_SOURCE,
      MPI_ANY_TAG, MPI_COMM_WORLD, &status);
    sender = status.MPI_SOURCE;
    anstype = status.MPI_TAG-1;
    for (j=0; j<M; j++)
      C[anstype][j] = ans[j];
    if (numsent<N) {
      for (j=0; j<L; j++)
        buffer[j] = A[numsent][j];
      MPI_Send(buffer, L, MPI_DOUBLE, sender,
        numsent+1, MPI_COMM_WORLD);
      numsent++;
    }
    else MPI_Send(buffer, 1, MPI_DOUBLE, sender,
      0, MPI_COMM_WORLD);
  }
} else { /* I am a slave */
  while (1) {
    MPI_Recv(buffer, L, MPI_DOUBLE, 0,
      MPI_ANY_TAG, MPI_COMM_WORLD, &status);
    if (status.MPI_TAG==0) break;
    row = status.MPI_TAG-1;
    for (i=0; i<M; i++) {
      ans[i] = 0.0;
      for (j=0; j<L; j++)
        ans[i] += buffer[j]*B[j][i];
    }
    MPI_Send(ans, M, MPI_DOUBLE, 0,
      row+1, MPI_COMM_WORLD);
  }
}
```

adapted from

Using MPI

by

William Gropp
Ewing Lusk
Anthony Skjellum

Why parallel numerical programs are difficult to get right

Why parallel numerical programs are difficult to get right

- usual reasons concurrent programming is difficult
  - parallelism adds complexity
  - nondeterminism
  - deadlocks
  - race conditions

Why parallel numerical programs are difficult to get right

- usual reasons concurrent programming is difficult
  - parallelism adds complexity
  - nondeterminism
  - deadlocks
  - race conditions
- additional sources of nondeterminism from MPI

Why parallel numerical programs are difficult to get right

- usual reasons concurrent programming is difficult
  - parallelism adds complexity
  - nondeterminism
  - deadlocks
  - race conditions
- additional sources of nondeterminism from MPI
- problem of test oracles
  - in scientific computation, often don't know correct result for a given test input, so can't tell if the observed result is correct
  - analytical solutions rarely exist
  - sequential program may only work on small test cases
  - floating-point arithmetic differs from real arithmetic

Current methods used to discover and correct problems in parallel numeric programs

Current methods used to discover and correct problems in parallel numeric programs

1. testing
   - only a tiny fraction of inputs can be tested
   - nondeterminism limits effectiveness
   - oracle problem

Current methods used to discover and correct problems in parallel numeric programs

1. testing
   - only a tiny fraction of inputs can be tested
   - nondeterminism limits effectiveness
   - oracle problem
2. parallel debuggers

Current methods used to discover and correct problems in parallel numeric programs

1. testing
   - only a tiny fraction of inputs can be tested
   - nondeterminism limits effectiveness
   - oracle problem
2. parallel debuggers
3. rewriting code in the hope that the problem will disappear
   - insertion of barriers

Our contribution

A method to verify parallel numerical programs.

Our contribution

A method to verify parallel numerical programs.

- verifies that the parallel program is functionally equivalent to a trusted sequential version

Our contribution

A method to verify parallel numerical programs.

- verifies that the parallel program is functionally equivalent to a trusted sequential version
  - for a given configuration

Our contribution

A method to verify parallel numerical programs.

- verifies that the parallel program is functionally equivalent to a trusted sequential version
  - for a given configuration
  - reduces the problem of verifying the correctness of a *parallel* numerical program to the problem of verifying the correctness of a *sequential* numerical program

Our contribution

A method to verify parallel numerical programs.

- verifies that the parallel program is functionally equivalent to a trusted sequential version
  - for a given configuration
  - reduces the problem of verifying the correctness of a *parallel* numerical program to the problem of verifying the correctness of a *sequential* numerical program
- uses symbolic execution to model floating-point computation

Our contribution

A method to verify parallel numerical programs.

- verifies that the parallel program is functionally equivalent to a trusted sequential version
  - for a given configuration
  - reduces the problem of verifying the correctness of a *parallel* numerical program to the problem of verifying the correctness of a *sequential* numerical program
- uses symbolic execution to model floating-point computation
- uses model checking
  - requires translation of program into input language of model checker

Our contribution

A method to verify parallel numerical programs.

- verifies that the parallel program is functionally equivalent to a trusted sequential version
  - for a given configuration
  - reduces the problem of verifying the correctness of a *parallel* numerical program to the problem of verifying the correctness of a *sequential* numerical program
- uses symbolic execution to model floating-point computation
- uses model checking
  - requires translation of program into input language of model checker
  - verifies equivalence over all executions
  - produces a trace if program is incorrect

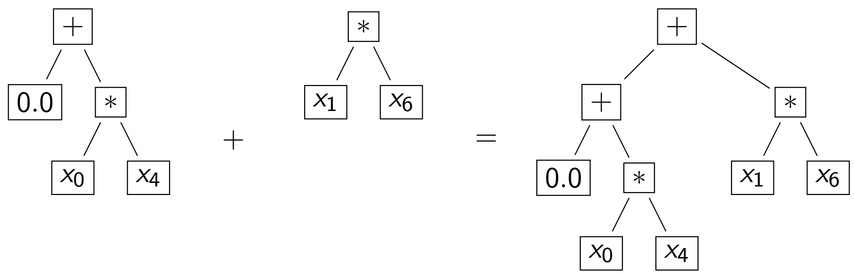
How do we model floating-point computation?

- one double-precision floating-point variable has 2^{64} possible states
- abstraction?

How do we model floating-point computation?

- one double-precision floating-point variable has 2^{64} possible states
- abstraction?

Input: symbolic constants $x_0, x_1, \ldots$
Output: symbolic expressions in the x_i



$$0.0 + (x_0 x_4) \quad + \quad x_1 x_6 \quad = \quad (0.0 + (x_0 x_4)) + x_1 x_6$$

How do we represent symbolic expressions?

Value numbering

- place all symbolic expressions in an expression table
  - every expression has a unique ID number

How do we represent symbolic expressions?

Value numbering

- place all symbolic expressions in an expression table
  - every expression has a unique ID number
- in the model. . .
  - replace all floating-point values with ID numbers

How do we represent symbolic expressions?

Value numbering

- place all symbolic expressions in an expression table
  - every expression has a unique ID number
- in the model...
  - replace all floating-point values with ID numbers
  - replace all floating-point operations with symbolic operations
    - to evaluate $x + y$:
    - is $x + y$ already in the table?
    - if *yes*, return its ID number
    - if *no*, create new table entry and return new ID number

i	e_i	interpretation

i	e_i	interpretation
0	$(\mathtt{L}, 0.0)$	0.0
1	$(\mathtt{L}, 1.0)$	1.0

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	$(\mathtt{L}, 0.0)$	0.0
1	$(\mathtt{L}, 1.0)$	1.0
2	$(\mathtt{X}, 0)$	x_0
3	$(\mathtt{X}, 1)$	x_1
4	$(\mathtt{X}, 2)$	x_2
5	$(\mathtt{X}, 3)$	x_3
6	$(\mathtt{X}, 4)$	x_4
7	$(\mathtt{X}, 5)$	x_5
8	$(\mathtt{X}, 6)$	x_6
9	$(\mathtt{X}, 7)$	x_7
10	$(*, 2, 6)$	$x_0 x_4$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	$(\mathtt{L}, 0.0)$	0.0
1	$(\mathtt{L}, 1.0)$	1.0
2	$(\mathtt{X}, 0)$	x_0
3	$(\mathtt{X}, 1)$	x_1
4	$(\mathtt{X}, 2)$	x_2
5	$(\mathtt{X}, 3)$	x_3
6	$(\mathtt{X}, 4)$	x_4
7	$(\mathtt{X}, 5)$	x_5
8	$(\mathtt{X}, 6)$	x_6
9	$(\mathtt{X}, 7)$	x_7
10	$(\mathtt{*}, 2, 6)$	$x_0 x_4$
11	$(\mathtt{+}, 0, 10)$	$0.0 + x_0 x_4$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \qquad C = \begin{pmatrix} 11 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \qquad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \qquad = \begin{pmatrix} 0.0 + x_0 x_4 & 0.0 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	$(\mathtt{L}, 0.0)$	0.0
1	$(\mathtt{L}, 1.0)$	1.0
2	$(\mathtt{X}, 0)$	x_0
3	$(\mathtt{X}, 1)$	x_1
4	$(\mathtt{X}, 2)$	x_2
5	$(\mathtt{X}, 3)$	x_3
6	$(\mathtt{X}, 4)$	x_4
7	$(\mathtt{X}, 5)$	x_5
8	$(\mathtt{X}, 6)$	x_6
9	$(\mathtt{X}, 7)$	x_7
10	$(*, 2, 6)$	$x_0 x_4$
11	$(+, 0, 10)$	$0.0 + x_0 x_4$
12	$(*, 3, 8)$	$x_1 x_6$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 11 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} 0.0 + x_0 x_4 & 0.0 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & 0.0 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \qquad C = \begin{pmatrix} 13 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \qquad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \qquad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & 0.0 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0 + x_0 x_5$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \qquad C = \begin{pmatrix} 13 & 15 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \qquad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \qquad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & 0.0 + x_0 x_5 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0 + x_0 x_5$
16	(*, 3, 9)	$x_1 x_7$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 15 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & 0.0 + x_0 x_5 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(∗, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(∗, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(∗, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0 + x_0 x_5$
16	(∗, 3, 9)	$x_1 x_7$
17	(+, 15, 16)	$(0.0 + x_0 x_5) + x_1 x_7$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 17 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & (0.0 + x_0 x_5) + x_1 x_7 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	x_0x_4
11	(+, 0, 10)	$0.0+x_0x_4$
12	(*, 3, 8)	x_1x_6

i	e_i	interpretation
13	(+, 11, 12)	$(0.0+x_0x_4)+x_1x_6$
14	(*, 2, 7)	x_0x_5
15	(+, 0, 14)	$0.0+x_0x_5$
16	(*, 3, 9)	x_1x_7
17	(+, 15, 16)	$(0.0+x_0x_5)+x_1x_7$
18	(*, 4, 6)	x_2x_4

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 17 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0+x_0x_4)+x_1x_6 & (0.0+x_0x_5)+x_1x_7 \\ 0.0 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	$(\mathtt{L}, 0.0)$	0.0
1	$(\mathtt{L}, 1.0)$	1.0
2	$(\mathtt{X}, 0)$	x_0
3	$(\mathtt{X}, 1)$	x_1
4	$(\mathtt{X}, 2)$	x_2
5	$(\mathtt{X}, 3)$	x_3
6	$(\mathtt{X}, 4)$	x_4
7	$(\mathtt{X}, 5)$	x_5
8	$(\mathtt{X}, 6)$	x_6
9	$(\mathtt{X}, 7)$	x_7
10	$(*, 2, 6)$	$x_0 x_4$
11	$(+, 0, 10)$	$0.0 + x_0 x_4$
12	$(*, 3, 8)$	$x_1 x_6$

i	e_i	interpretation
13	$(+, 11, 12)$	$(0.0 + x_0 x_4) + x_1 x_6$
14	$(*, 2, 7)$	$x_0 x_5$
15	$(+, 0, 14)$	$0.0 + x_0 x_5$
16	$(*, 3, 9)$	$x_1 x_7$
17	$(+, 15, 16)$	$(0.0 + x_0 x_5) + x_1 x_7$
18	$(*, 4, 6)$	$x_2 x_4$
19	$(+, 0, 12)$	$0.0 + x_2 x_4$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \qquad C = \begin{pmatrix} 13 & 17 \\ 19 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \qquad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \qquad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & (0.0 + x_0 x_5) + x_1 x_7 \\ 0.0 + x_2 x_4 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0 + x_0 x_5$
16	(*, 3, 9)	$x_1 x_7$
17	(+, 15, 16)	$(0.0 + x_0 x_5) + x_1 x_7$
18	(*, 4, 6)	$x_2 x_4$
19	(+, 0, 12)	$0.0 + x_2 x_4$
20	(*, 5, 8)	$x_3 x_6$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 17 \\ 19 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & (0.0 + x_0 x_5) + x_1 x_7 \\ 0.0 + x_2 x_4 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0 + x_0 x_5$
16	(*, 3, 9)	$x_1 x_7$
17	(+, 15, 16)	$(0.0 + x_0 x_5) + x_1 x_7$
18	(*, 4, 6)	$x_2 x_4$
19	(+, 0, 12)	$0.0 + x_2 x_4$
20	(*, 5, 8)	$x_3 x_6$
21	(+, 19, 20)	$(0.0 + x_2 x_4) + x_3 x_6$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 17 \\ 21 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & (0.0 + x_0 x_5) + x_1 x_7 \\ (0.0 + x_2 x_4) + x_3 x_6 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0 + x_0 x_5$
16	(*, 3, 9)	$x_1 x_7$
17	(+, 15, 16)	$(0.0 + x_0 x_5) + x_1 x_7$
18	(*, 4, 6)	$x_2 x_4$
19	(+, 0, 12)	$0.0 + x_2 x_4$
20	(*, 5, 8)	$x_3 x_6$
21	(+, 19, 20)	$(0.0 + x_2 x_4) + x_3 x_6$
22	(*, 4, 7)	$x_2 x_5$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \qquad C = \begin{pmatrix} 13 & 17 \\ 21 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \qquad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \qquad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & (0.0 + x_0 x_5) + x_1 x_7 \\ (0.0 + x_2 x_4) + x_3 x_6 & 0.0 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0 + x_0 x_5$
16	(*, 3, 9)	$x_1 x_7$
17	(+, 15, 16)	$(0.0 + x_0 x_5) + x_1 x_7$
18	(*, 4, 6)	$x_2 x_4$
19	(+, 0, 12)	$0.0 + x_2 x_4$
20	(*, 5, 8)	$x_3 x_6$
21	(+, 19, 20)	$(0.0 + x_2 x_4) + x_3 x_6$
22	(*, 4, 7)	$x_2 x_5$
23	(+, 0, 22)	$0.0 + x_2 x_5$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 17 \\ 21 & 23 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & (0.0 + x_0 x_5) + x_1 x_7 \\ (0.0 + x_2 x_4) + x_3 x_6 & 0.0 + x_2 x_5 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0+x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0+x_0 x_4)+x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0+x_0 x_5$
16	(*, 3, 9)	$x_1 x_7$
17	(+, 15, 16)	$(0.0+x_0 x_5)+x_1 x_7$
18	(*, 4, 6)	$x_2 x_4$
19	(+, 0, 12)	$0.0+x_2 x_4$
20	(*, 5, 8)	$x_3 x_6$
21	(+, 19, 20)	$(0.0+x_2 x_4)+x_3 x_6$
22	(*, 4, 7)	$x_2 x_5$
23	(+, 0, 22)	$0.0+x_2 x_5$
24	(*, 5, 9)	$x_3 x_7$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 17 \\ 21 & 23 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0+x_0 x_4)+x_1 x_6 & (0.0+x_0 x_5)+x_1 x_7 \\ (0.0+x_2 x_4)+x_3 x_6 & 0.0+x_2 x_5 \end{pmatrix}$$

i	e_i	interpretation
0	(L, 0.0)	0.0
1	(L, 1.0)	1.0
2	(X, 0)	x_0
3	(X, 1)	x_1
4	(X, 2)	x_2
5	(X, 3)	x_3
6	(X, 4)	x_4
7	(X, 5)	x_5
8	(X, 6)	x_6
9	(X, 7)	x_7
10	(*, 2, 6)	$x_0 x_4$
11	(+, 0, 10)	$0.0 + x_0 x_4$
12	(*, 3, 8)	$x_1 x_6$

i	e_i	interpretation
13	(+, 11, 12)	$(0.0 + x_0 x_4) + x_1 x_6$
14	(*, 2, 7)	$x_0 x_5$
15	(+, 0, 14)	$0.0 + x_0 x_5$
16	(*, 3, 9)	$x_1 x_7$
17	(+, 15, 16)	$(0.0 + x_0 x_5) + x_1 x_7$
18	(*, 4, 6)	$x_2 x_4$
19	(+, 0, 12)	$0.0 + x_2 x_4$
20	(*, 5, 8)	$x_3 x_6$
21	(+, 19, 20)	$(0.0 + x_2 x_4) + x_3 x_6$
22	(*, 4, 7)	$x_2 x_5$
23	(+, 0, 22)	$0.0 + x_2 x_5$
24	(*, 5, 9)	$x_3 x_7$
25	(+, 23, 24)	$(0.0 + x_2 x_5) + x_3 x_7$

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 13 & 17 \\ 21 & 25 \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad = \begin{pmatrix} x_4 & x_5 \\ x_6 & x_7 \end{pmatrix} \quad = \begin{pmatrix} (0.0 + x_0 x_4) + x_1 x_6 & (0.0 + x_0 x_5) + x_1 x_7 \\ (0.0 + x_2 x_4) + x_3 x_6 & (0.0 + x_2 x_5) + x_3 x_7 \end{pmatrix}$$

The path correspondence problem

- the programs may contain branches on expressions that involve the symbolic variables
  - **if** $(x_0 \neq 0)$ $\{\ldots\}$ **else** $\{\ldots\}$

The path correspondence problem

- the programs may contain branches on expressions that involve the symbolic variables
  - **if** $(x_0 \neq 0)$ $\{\ldots\}$ **else** $\{\ldots\}$
- only want to compare the result of an execution path in the parallel program to the result of a *corresponding* path in the sequential program

Path conditions and domains

- enumerate all paths through the sequential program
  - keeping track of the path condition for each path

$$\mathbf{y} = \begin{cases} f_1(\mathbf{x}) & \text{if } p_1(\mathbf{x}) \\ f_2(\mathbf{x}) & \text{if } p_2(\mathbf{x}) \\ \vdots & \vdots \\ f_n(\mathbf{x}) & \text{if } p_n(\mathbf{x}) \end{cases}$$

Path conditions and domains

- enumerate all paths through the sequential program
  - keeping track of the path condition for each path

$$\mathbf{y} = \begin{cases} f_1(\mathbf{x}) & \text{if } p_1(\mathbf{x}) \\ f_2(\mathbf{x}) & \text{if } p_2(\mathbf{x}) \\ \vdots & \vdots \\ f_n(\mathbf{x}) & \text{if } p_n(\mathbf{x}) \end{cases}$$

- each p_i determines a path domain $D_i = \{\mathbf{x} \mid p_i(\mathbf{x})\}$
- $D_i \cap D_j = \emptyset$ if $i \neq j$
- $\cup_i D_i$ is the whole input space

Path conditions and domains

- enumerate all paths through the sequential program
  - keeping track of the path condition for each path

$$\mathbf{y} = \begin{cases} f_1(\mathbf{x}) & \text{if } p_1(\mathbf{x}) \\ f_2(\mathbf{x}) & \text{if } p_2(\mathbf{x}) \\ \vdots & \vdots \\ f_n(\mathbf{x}) & \text{if } p_n(\mathbf{x}) \end{cases}$$

- each p_i determines a path domain $D_i = \{\mathbf{x} \mid p_i(\mathbf{x})\}$
- $D_i \cap D_j = \emptyset$ if $i \neq j$
- $\cup_i D_i$ is the whole input space

Solution to path correspondence problem:

1. discover path conditions/domains automatically
2. for each domain D_i: compare symbolic results of sequential and parallel programs for all inputs in D_i

Modeling conditional statements

To model the statement **if** $(x_0 \neq 0)\ \{\ldots\}$ **else** $\{\ldots\}$

$p \leftarrow$ **true**; /* path condition */
$\vdots$
$b \leftarrow \mu(p, x_0 \neq 0)$;
if $(b = -1)$ {
  if $(choose())$ {
    $b \leftarrow 1$; $p \leftarrow p \wedge (x_0 \neq 0)$;
  } **else** {
    $b \leftarrow 0$; $p \leftarrow p \wedge (x_0 = 0)$;
  }
}
if $(b = 1)$ { ... } **else** { ... }

$$\mu(p, q) = \begin{cases} 1 & \text{if } p \Rightarrow q \\ 0 & \text{if } p \Rightarrow \neg q \\ -1 & \text{if don't know} \end{cases}$$

for boolean-valued symbolic expressions p, q

The method

1. construct symbolic model M_{seq} of sequential program
   - input: $\mathbf{x}$, output: $\mathbf{y}$, path condition: p
2. construct symbolic model M_{par} of parallel program
   - input: $\mathbf{x}$, output: $\mathbf{y}'$, path condition: p
   - using same symbolic table
3. create composite model M:
   - $p \leftarrow \textbf{true}$; M_{seq}; M_{par}; $\texttt{assert(}\mathbf{y} = \mathbf{y}'\texttt{);}$
4. use model checker to verify the assertion in M can never be violated

The method

1. construct symbolic model M_{seq} of sequential program
   - input: $\mathbf{x}$, output: $\mathbf{y}$, path condition: p
2. construct symbolic model M_{par} of parallel program
   - input: $\mathbf{x}$, output: $\mathbf{y}'$, path condition: p
   - using same symbolic table
3. create composite model M:
   - $p \leftarrow \textbf{true}$; M_{seq}; M_{par}; $\texttt{assert}(\mathbf{y} = \mathbf{y}')\texttt{;}$
4. use model checker to verify the assertion in M can never be violated

The model checker returns either

- Yes: the property holds, or

The method

1. construct symbolic model M_{seq} of sequential program
   - input: $\mathbf{x}$, output: $\mathbf{y}$, path condition: p
2. construct symbolic model M_{par} of parallel program
   - input: $\mathbf{x}$, output: $\mathbf{y}'$, path condition: p
   - using same symbolic table
3. create composite model M:
   - $p \leftarrow \textbf{true}$; M_{seq}; M_{par}; `assert(`$\mathbf{y} = \mathbf{y}'$`);`
4. use model checker to verify the assertion in M can never be violated

The model checker returns either

- Yes: the property holds, or
- No + counterexample:
  - a trace through M_{seq}
  - a trace through M_{par}
  - the values of p, $\mathbf{y}$, and $\mathbf{y}'$

Numerical Issues

different symbolic expressions are equivalent over real numbers

- example: $((x_3 + x_1) + x_2) + x_0$ and $((x_0 + x_1) + x_2) + x_3$

Numerical Issues

different symbolic expressions are equivalent over real numbers

- example: $((x_3 + x_1) + x_2) + x_0$ and $((x_0 + x_1) + x_2) + x_3$
- solution: provide different modes
    1. Herbrand
        - exact equality of expressions

Numerical Issues

different symbolic expressions are equivalent over real numbers

- example: $((x_3 + x_1) + x_2) + x_0$ and $((x_0 + x_1) + x_2) + x_3$
- solution: provide different modes
  1. Herbrand
     - exact equality of expressions
  2. IEEE
     - $x + y = y + x$
     - $1x = x = x1$
     - $x + 0 = x = 0 + x$

Numerical Issues

different symbolic expressions are equivalent over real numbers

- example: $((x_3 + x_1) + x_2) + x_0$ and $((x_0 + x_1) + x_2) + x_3$
- solution: provide different modes
  1. Herbrand
     - exact equality of expressions
  2. IEEE
     - $x + y = y + x$
     - $1x = x = x1$
     - $x + 0 = x = 0 + x$
  3. Real
     - all of IEEE
     - $(x + y) + z = x + (y + z)$
     - $(xy)z = x(yz)$
     - ...

Preliminary experimental results

- implemented as extension to SPIN
- wrote our own simple symbolic algebra library and light-weight theorem-prover
- found bug in one example (Jacobi iteration)

	matrix multiplication	Gaussian elimination	Jacobi iteration	Monte Carlo
processes	6	6	17	9
path domains	1	13327	4	4
symbolic expressions	2202	247656	8239	1232
input vector size	200	36	1333	99
output vector size	100	36	36	1
states (10^3)	4443	16114	6295	3112
memory (MB)	217	801	362	279
time (s)	506	3224	9846	738

Related Work

1. Ball and Rajamani. *Automatically validating temporal safety properties of interfaces*. SPIN 2001.
2. Khurshid, Păsăreanu, and Visser. *Generalized symbolic execution for model checking and testing*. TACAS 2003.
3. Păsăreanu and Visser. *Verification of Java programs using symbolic execution and invariant generation*. SPIN 2004.
4. Elmas, Tasiran, and Qadeer. *VYRD: verifying concurrent programs by runtime refinement-violation detection*. PLDI 2005.

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program. . .
  - . . .over all inputs

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program. . .
  - . . .over all inputs
  - . . .over all possible executions

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program. . .
  - . . .over all inputs
  - . . .over all possible executions
- can be largely automated

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program. . .
  - . . .over all inputs
  - . . .over all possible executions
- can be largely automated
- conservative

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program...
  - ...over all inputs
  - ...over all possible executions
- can be largely automated
- conservative
- produces detailed counterexample if equivalence cannot be verified

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program...
  - ...over all inputs
  - ...over all possible executions
- can be largely automated
- conservative
- produces detailed counterexample if equivalence cannot be verified

Weaknesses of method

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program. . .
  - . . .over all inputs
  - . . .over all possible executions
- can be largely automated
- conservative
- produces detailed counterexample if equivalence cannot be verified

Weaknesses of method

- models are constructed by hand (for now)

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program...
  - ...over all inputs
  - ...over all possible executions
- can be largely automated
- conservative
- produces detailed counterexample if equivalence cannot be verified

Weaknesses of method

- models are constructed by hand (for now)
- state explosion
  - often requires small bounds on configuration

Conclusion

Strengths of method

- can establish the correctness of a parallel numerical program...
  - ...over all inputs
  - ...over all possible executions
- can be largely automated
- conservative
- produces detailed counterexample if equivalence cannot be verified

Weaknesses of method

- models are constructed by hand (for now)
- state explosion
  - often requires small bounds on configuration
- possibility of spurious error report
  - precision depends upon close correspondence between numerical operations in sequential and parallel programs

Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \\ & \\ & \\ & \end{bmatrix}$$

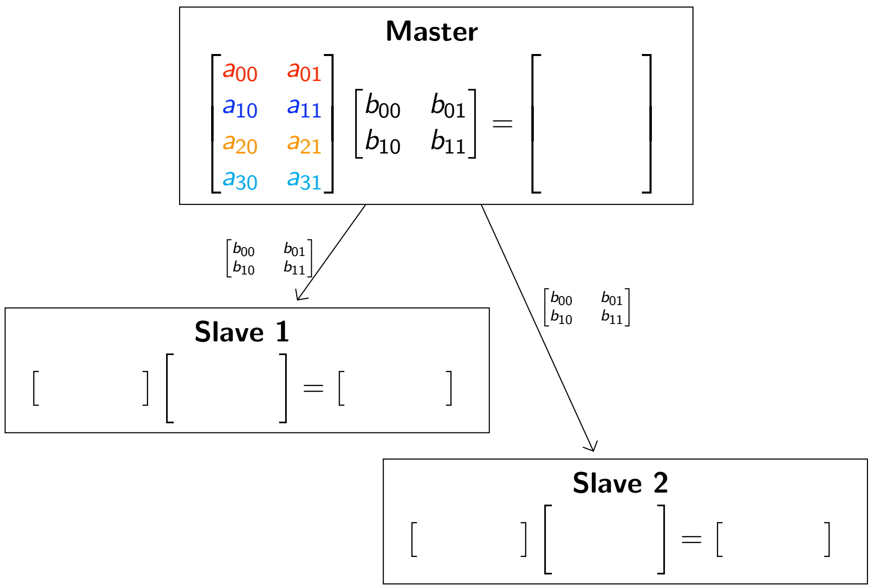
Slave 1

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} \quad & \quad \\ & \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} \quad & \quad \\ & \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication



Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \\ & \\ & \\ & \end{bmatrix}$$

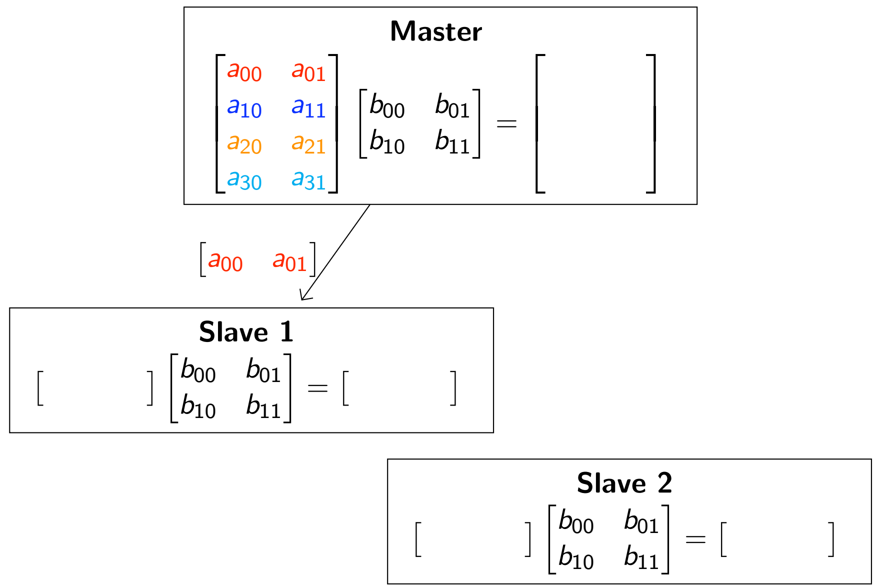
Slave 1

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication



Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}$$

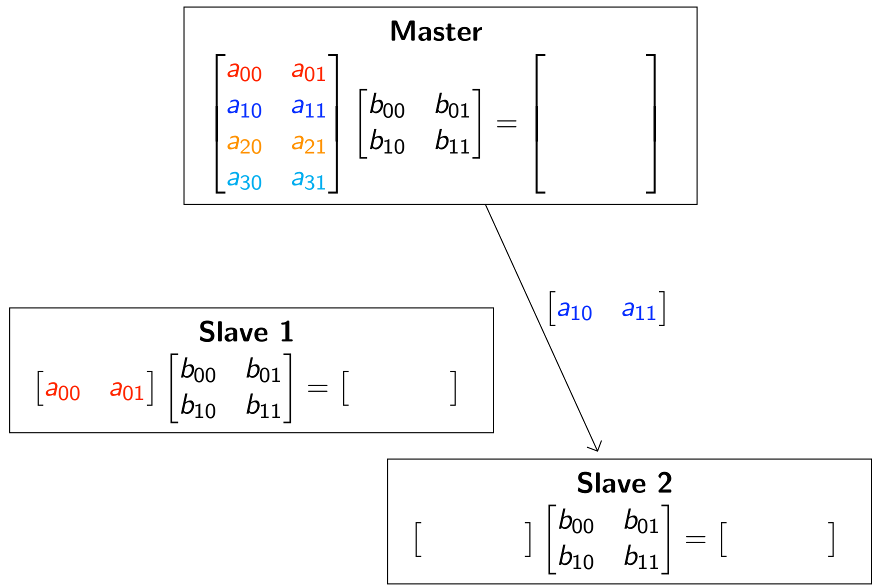
Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} & \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Master-slave matrix multiplication



Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}$$

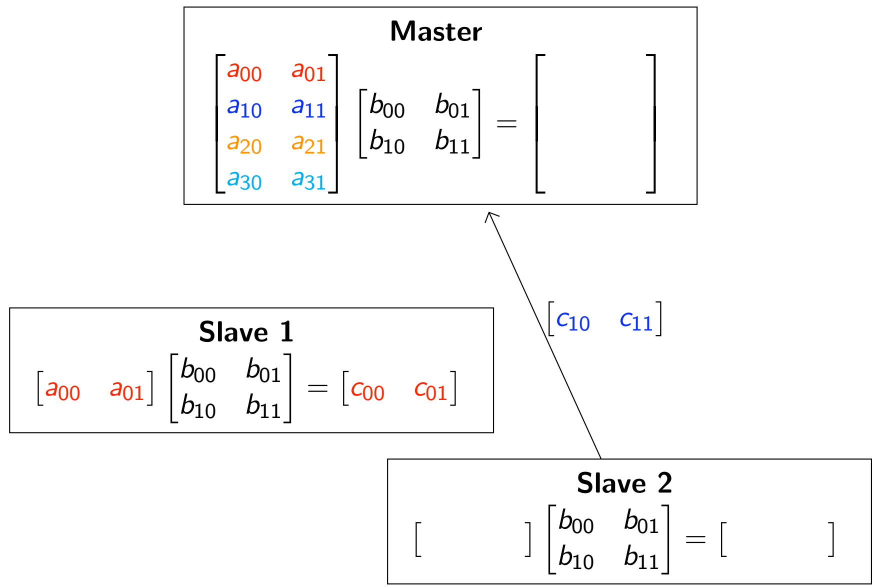
Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{10} & c_{11} \end{bmatrix}$$

Master-slave matrix multiplication



Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \\ c_{10} & c_{11} \\ & \\ & \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{10} & c_{11} \end{bmatrix}$$

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \\ c_{10} & c_{11} \\ & \\ & \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \\ c_{10} & c_{11} \\ & \\ & \end{bmatrix}$$

$$\begin{bmatrix} c_{01} & c_{11} \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ & \\ & \end{bmatrix}$$

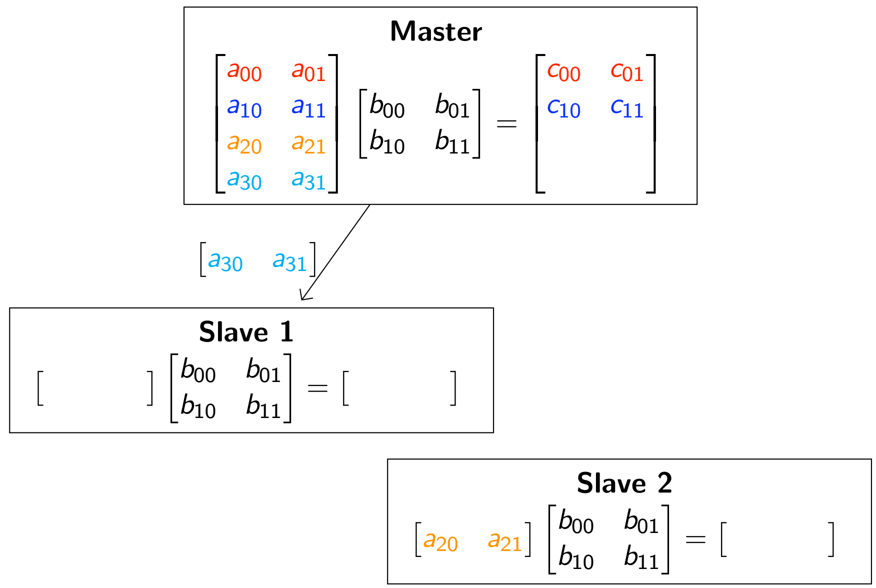
Slave 1

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication



Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ & \\ & \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ & \\ & \end{bmatrix}$$

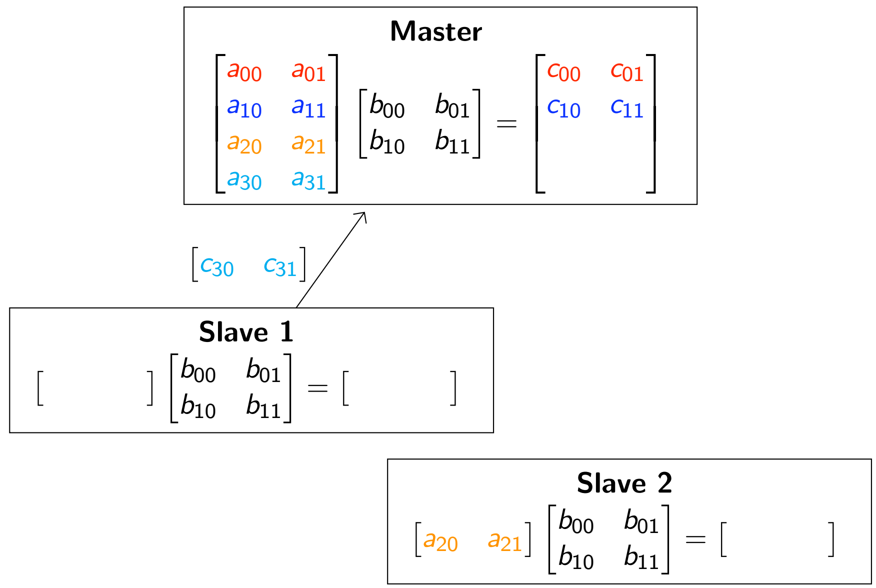
Slave 1

$$\begin{bmatrix} a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{30} & c_{31} \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication



Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ & \\ c_{30} & c_{31} \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} & \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ & \\ c_{30} & c_{31} \end{bmatrix}$$

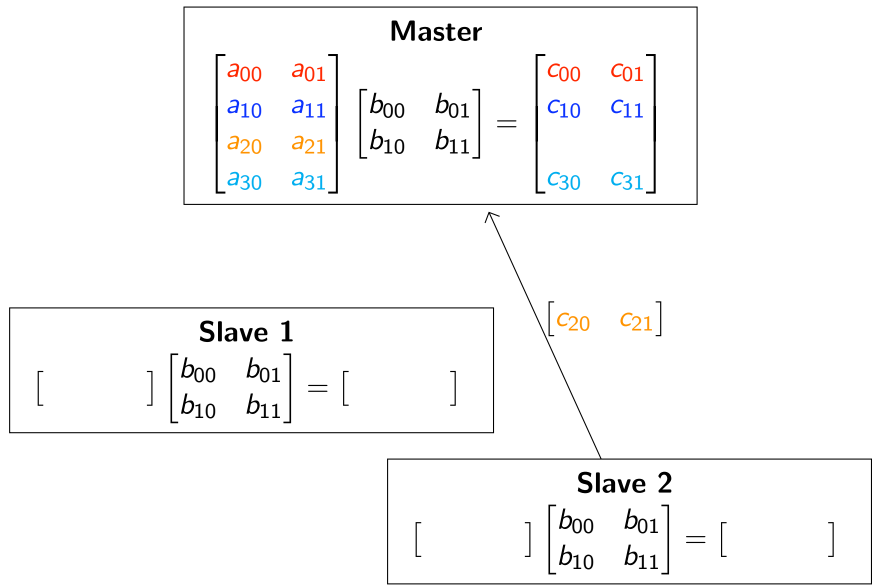
Slave 1

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{20} & c_{21} \end{bmatrix}$$

Master-slave matrix multiplication



Master-slave matrix multiplication

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ c_{20} & c_{21} \\ c_{30} & c_{31} \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication: execution 2

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

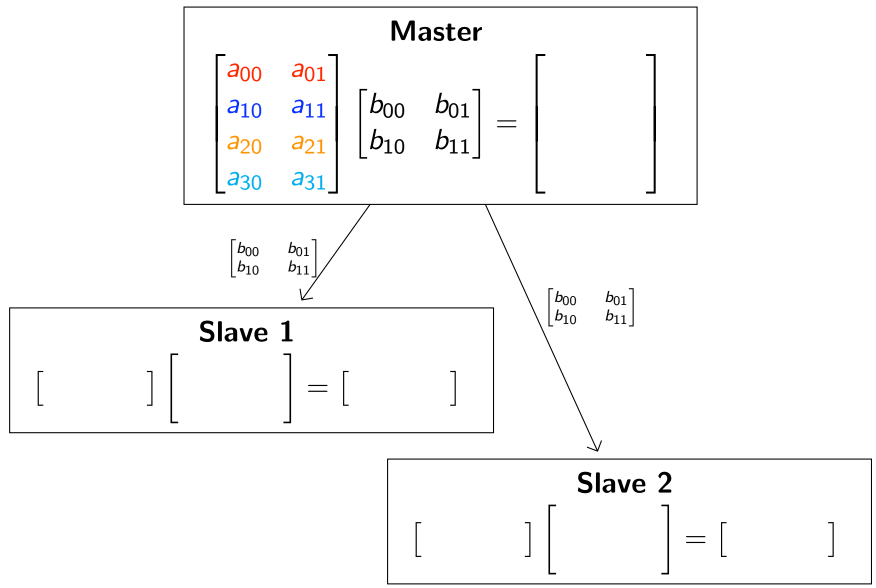
Slave 1

$$\begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} \quad \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} \quad \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix}$$

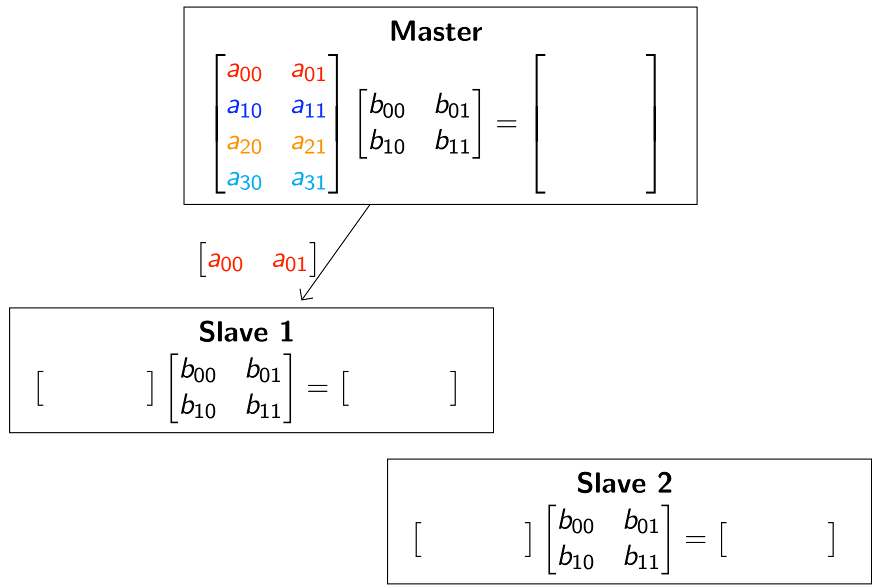
Slave 1

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \\ & \\ & \\ & \end{bmatrix}$$

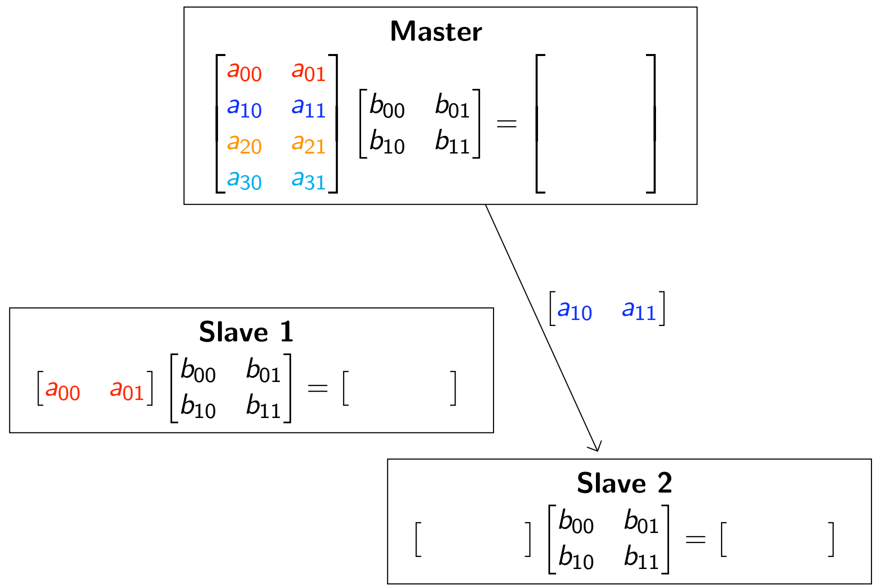
Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} & \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \\ & \\ & \\ & \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

Master-slave matrix multiplication: execution 2

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \\ & \\ & \\ & \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} a_{00} & a_{01} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{10} & c_{11} \end{bmatrix}$$

Master-slave matrix multiplication: execution 2

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}$$

$$\begin{bmatrix} c_{01} & c_{11} \end{bmatrix}$$

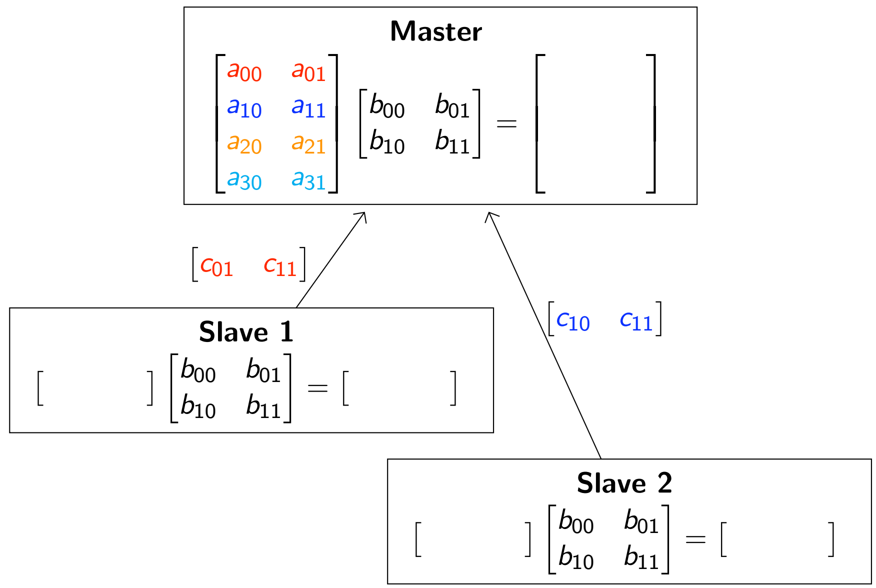
Slave 1

$$\begin{bmatrix} & \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \end{bmatrix}$$

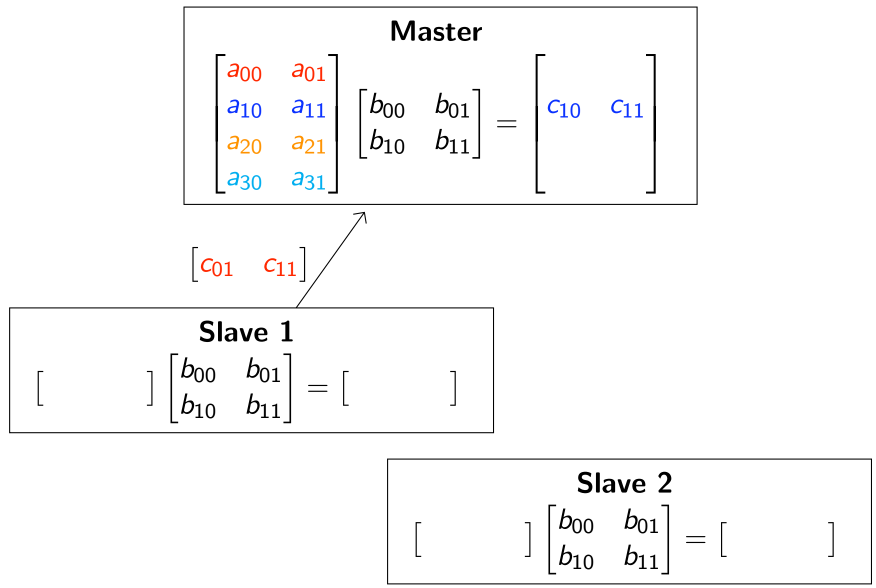
Slave 2

$$\begin{bmatrix} a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{10} & c_{11} \end{bmatrix}$$

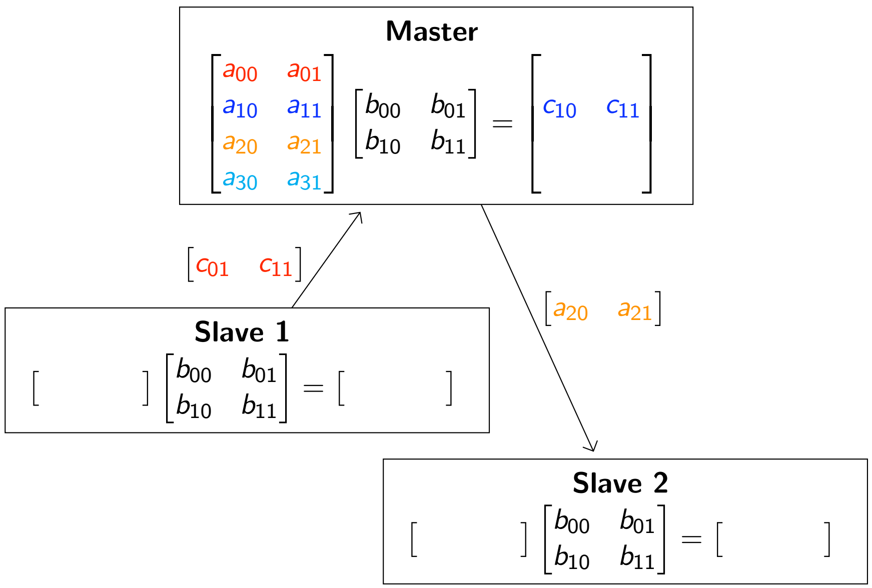
Master-slave matrix multiplication: execution 2



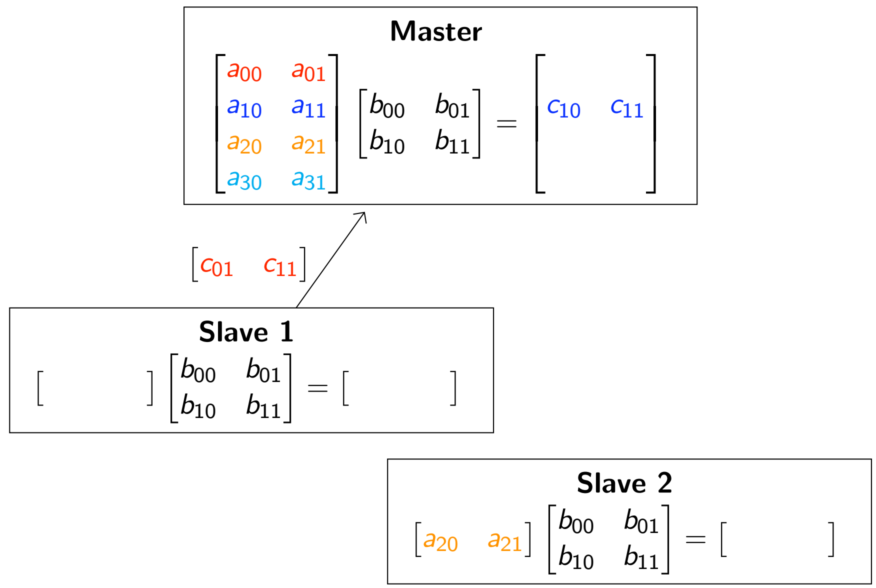
Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} & \\ c_{10} & c_{11} \\ & \\ & \end{bmatrix}$$

$$\begin{bmatrix} c_{01} & c_{11} \end{bmatrix}$$

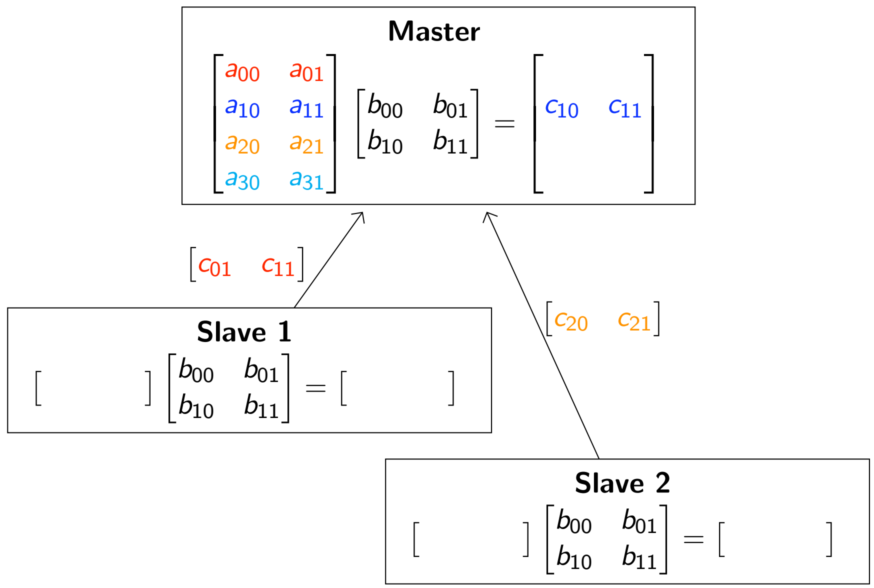
Slave 1

$$\begin{bmatrix} \quad & \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

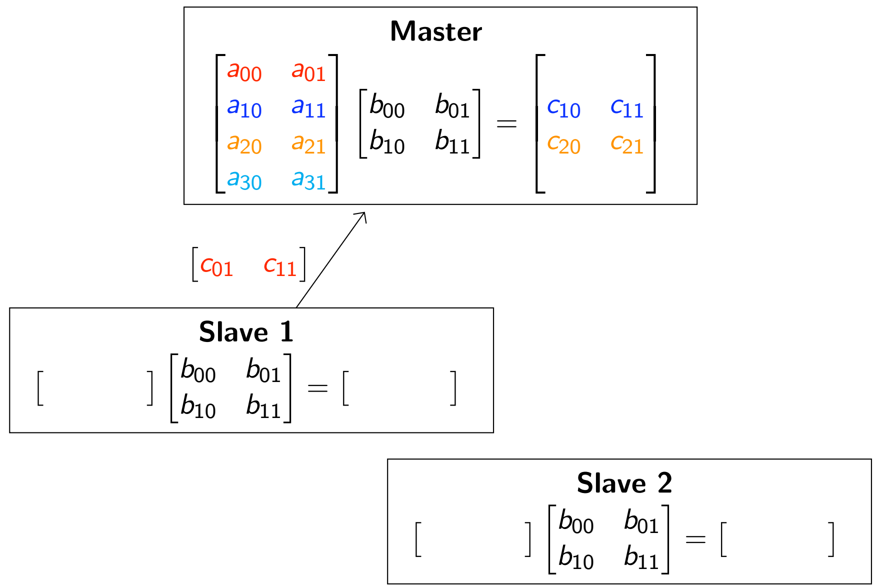
Slave 2

$$\begin{bmatrix} a_{20} & a_{21} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{20} & c_{21} \end{bmatrix}$$

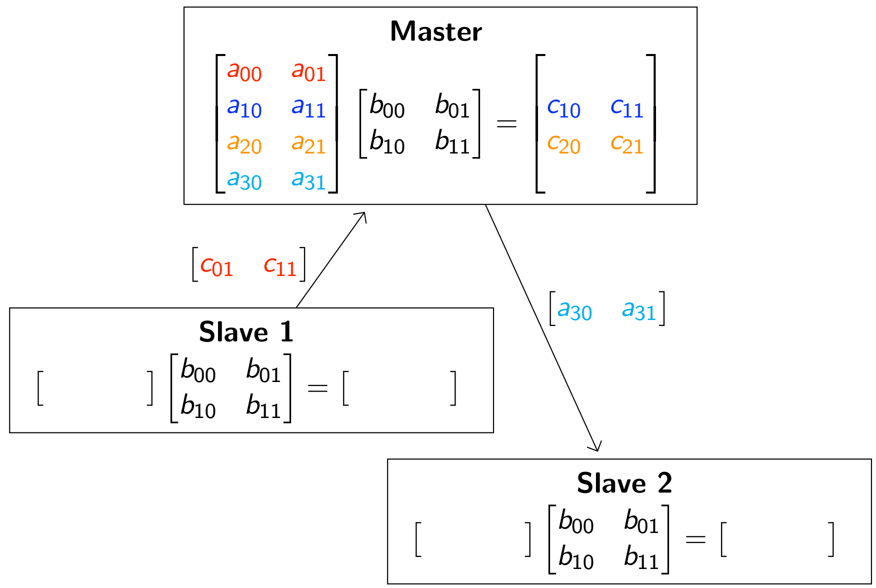
Master-slave matrix multiplication: execution 2



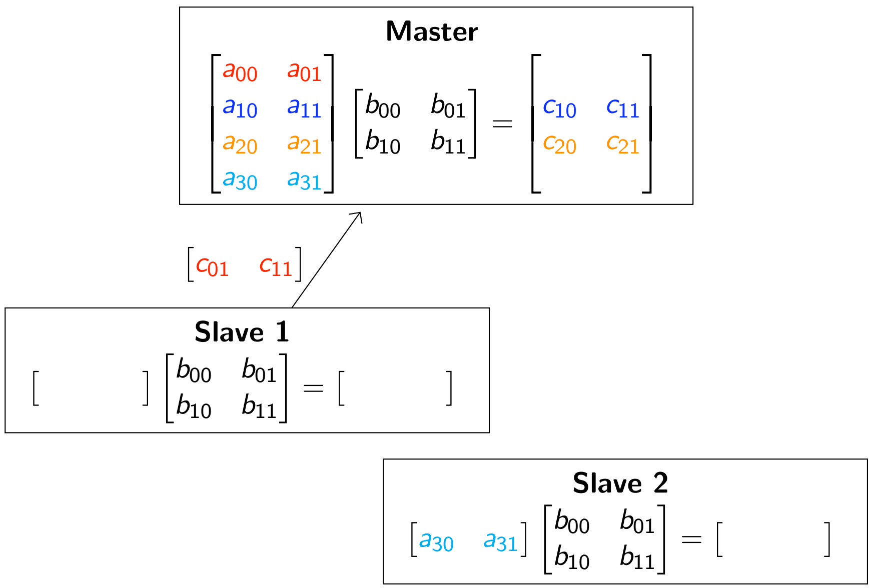
Master-slave matrix multiplication: execution 2



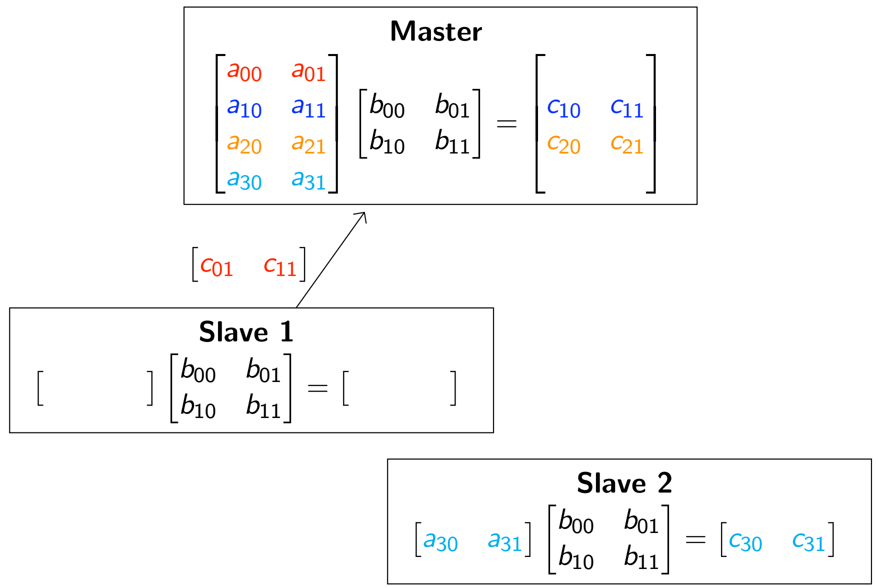
Master-slave matrix multiplication: execution 2



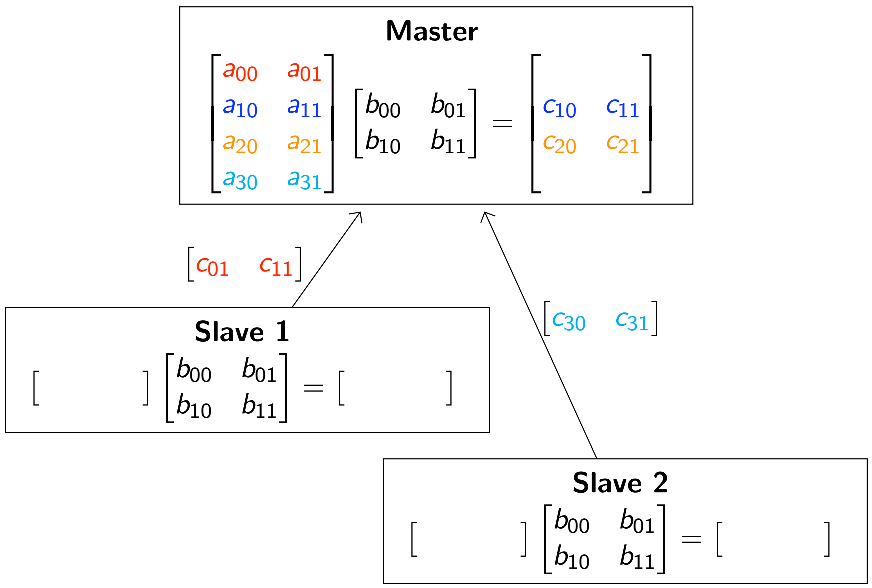
Master-slave matrix multiplication: execution 2



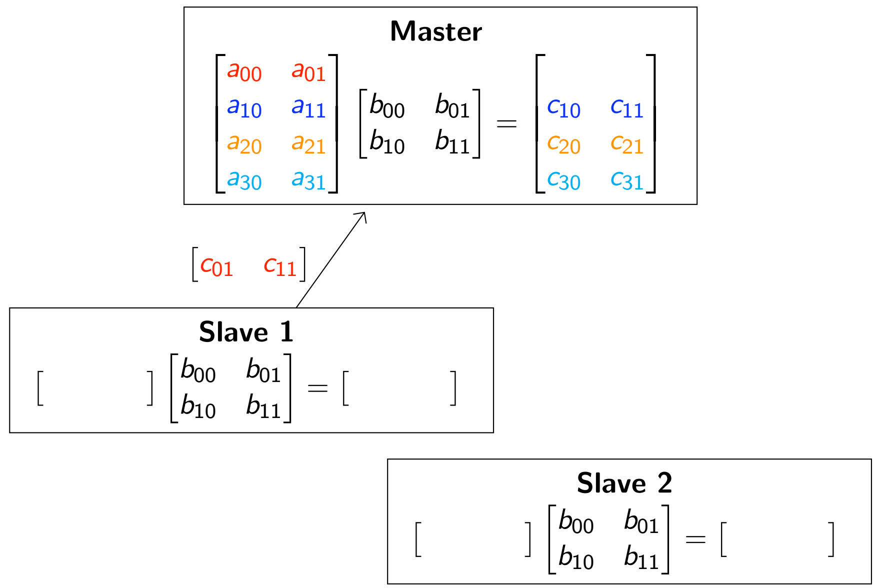
Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2



Master-slave matrix multiplication: execution 2

Master

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \\ a_{30} & a_{31} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ c_{20} & c_{21} \\ c_{30} & c_{31} \end{bmatrix}$$

Slave 1

$$\begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

Slave 2

$$\begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

Gaussian elimination

Step 1 Locate the leftmost column of A that does not consist entirely of zeros, if one exists. The top nonzero entry of this column is the pivot.

Step 2 Interchange the top row with the pivot row, if necessary, so that the entry at the top of the column found in Step 1 is nonzero.

Step 3 Divide the top row by pivot in order to introduce a leading 1.

Step 4 Add suitable multiples of the top row to all other rows so that all entries above and below the leading 1 become zero.

Repeat.

Gaussian elimination

transforms a matrix to its reduced row-echelon form:

$$\mathbf{x} = \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad \longrightarrow \quad \mathbf{y} = \begin{pmatrix} y_0 & y_1 \\ y_2 & y_3 \end{pmatrix}$$

Gaussian elimination

transforms a matrix to its reduced row-echelon form:

$$\mathbf{x} = \begin{pmatrix} x_0 & x_1 \\ x_2 & x_3 \end{pmatrix} \quad \longrightarrow \quad \mathbf{y} = \begin{pmatrix} y_0 & y_1 \\ y_2 & y_3 \end{pmatrix}$$

$$\mathbf{y} = \begin{cases} \left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right) & \text{if } x_0 = 0 \land x_2 = 0 \land x_1 = 0 \land x_3 = 0 \\ \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right) & \text{if } x_0 = 0 \land x_2 = 0 \land x_1 = 0 \land x_3 \neq 0 \\ \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right) & \text{if } x_0 = 0 \land x_2 = 0 \land x_1 \neq 0 \\ \left(\begin{smallmatrix} 1 & x_3/x_2 \\ 0 & 0 \end{smallmatrix}\right) & \text{if } x_0 = 0 \land x_2 \neq 0 \land x_1 = 0 \\ \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) & \text{if } x_0 = 0 \land x_2 \neq 0 \land x_1 \neq 0 \\ \left(\begin{smallmatrix} 1 & x_1/x_0 \\ 0 & 0 \end{smallmatrix}\right) & \text{if } x_0 \neq 0 \land x_3 - x_2(x_1/x_0) = 0 \\ \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) & \text{if } x_0 \neq 0 \land x_3 - x_2(x_1/x_0) \neq 0 \end{cases}$$