Communication Lower Bounds for Programs with Affine Dependences

Michael Christ, James Demmel, Nicholas Knight, Thomas Scanlon, Katherine Yelick
University of California, Berkeley

SIAM PP14
February 19, 2014

We acknowledge funding from Microsoft (award #024263) and Intel (award #024894), and matching funding by UC Discovery (award #DIG07-10227), with additional support from ParLab affiliates National Instruments, Nokia, NVIDIA, Oracle, and Samsung; from DARPA (award #HR0011-12-2-0016), the Center for Future Architecture Research, a member of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and ASPIRE Lab industrial sponsors and affiliates Intel, Google, Nokia, NVIDIA, and Oracle; from MathWorks; and from the US DOE (grants DE-SC0003959, DE-SC0004938, DE-SC0005136, DE-SC0008700, DE-AC02-05CH11231, DE-FC02-06ER25753, and DE-FC02-07ER25799) and NSF (grant DMS-0901569).
Communication is costly

*Communication* means moving data:

- Serial communication: moving data within memory hierarchy
- Parallel communication: moving data within network

- Communication cost often dominates runtime and energy
  ⇒ avoiding communication can save time and energy
- Seek *lower bounds*, and *optimal algorithms* that attain them
- This talk: optimizing *loop nests with affine dependences*
Algorithmic Costs

• Every cycle, a processor is computing, communicating, or idle; runtime $T$ modeled by

$$\max\{T_{\text{comp}}, T_{\text{comm}}\} \leq T - T_{\text{idle}} \leq T_{\text{comp}} + T_{\text{comm}}.$$  

(In parallel, runtime is the maximum over all processors’ $T$s.)

• Given a class of computations, we wish to find the one that minimizes $T$ on a particular machine.

• For our application (loop nest optimization):

  $T_{\text{comm}}$ reduce by improving data locality

  $T_{\text{comp}}, T_{\text{idle}}$ ignore; assume that they don’t worsen when optimizing for data locality, or that $T_{\text{comm}}$ is dominant cost
Communication Model

Definition

A sequential or parallel machine has a memory \((x_1, x_2, \ldots)\) of cells, addressed by 1, 2, \ldots, and each storing a value \(x_i \in D \cup \{e\}\), where \(D\) is a given domain and \(e \notin D\) is a sentinel value indicating that a cell is empty. The function \(\ell : \{1, 2, \ldots\} \rightarrow [0, \infty)\) denotes the (best-case) access cost of each memory address.

For example,

- 2-level memory, given \(\alpha, M\): \(\ell(a) = \alpha 1_{\{M+1,M+2,\ldots\}}(a)\)
- \((L + 1)\)-level memory \((L \geq 1)\), given \(((\alpha_1, M_1), \ldots, (\alpha_L, M_L))\):

\[
\ell(a) = \sum_{i=1}^{L} \left( \alpha_i - \sum_{j=1}^{i-1} \alpha_j \right) 1_{\{M_i+1,M_i+2,\ldots\}}(a)
\]

- ‘Ideal’ hierarchy: \(\ell(a) = \lceil \log a \rceil\)
- \(d\)-dimensional memory: \(\ell(a) = \Theta(a^{1/d})\)
- Parallel case: each processor has own \(\ell\).
Communication Cost

Definition

The *communication cost* of a sequence $a_1, a_2, \ldots$ of accesses is

$$T_{\text{comm}} \geq \sum_i \ell(a_i);$$

this may not be tight when the actual access costs vary, e.g., due to congestion/routing.

(Note: no overlap of different accesses’ costs; see next slide).

Definition

With respect to a given sequence $(a_1, a_2, \ldots)$ of accesses and a given $M \in \mathbb{N}$, we define the *communication volume* $Q_M = |\{i : a_i > M\}|$, representing the number of words moved between a slow memory of unbounded capacity and a fast memory of size $M$.

- We can write $\sum_i \ell(a_i) = \ell(1)Q_0 + \sum_{n=1}^{\infty} Q_n \Delta \ell(n)$, where $\Delta$ denotes the forward difference.
  - E.g., if $\ell(a) = \alpha 1_{\{M+1, M+2, \ldots\}}(a)$, then $T_{\text{comm}} \geq \alpha Q_M$.
- Supposing $\ell$ is nondecreasing (WLOG), a lower bound on $Q_M$ (for all $M \in \mathbb{N}$) can be used to bound $T_{\text{comm}}$ below.
In many cases, our analysis can be used to obtain lower bounds for machines that exploit concurrency in the memory system with blocked/pipelined transfers:

- Sequence of messages \((b_1, \ldots, b_S)\), each \(b_i = \{a_{i,1}, \ldots, a_{i,m_i}\}\) collecting \(m_i\) distinct accesses.

- \(\ell_{\text{blk}}(b_i)\) is a function of \(\{a_{i,1}, \ldots, a_{i,m_i}\}\); \(T_{\text{comm}} \geq \sum_i \ell_{\text{blk}}(b_i)\)

  - E.g., for a 2-level memory with \(\ell(a) = \alpha 1_{\{M+1, M+2, \ldots\}}(a)\), parameterized by \(\alpha, M\), maximum blocksize \(B \geq m_i\), maximum bandwidth \(1/\beta\), and
    \[\ell_{\text{blk}}(b_i) = \max\{\ell(a_{i,j}) : j \in \{1, \ldots, m_i\}\} + \beta|\{j : a_{i,j} > M\}|,\] then
    \[T_{\text{comm}} \geq \alpha Q_M/B + \beta Q_M,\] the latency cost plus the bandwidth cost.

Blocking increases the granularity of communication; in parallel, may decrease the frequency of synchronization:

- Possibly reduces latency cost (component of \(T_{\text{comm}}\));

- Tradeoffs: blocking may increase bandwidth cost, \(T_{\text{comp}}, T_{\text{idle}}\), for some processors (analyze critical paths).
Computation Directed Acyclic Graphs (CDAGs)

We model a computation as a directed acyclic graph (DAG):

**Definition**

A *computation* DAG (CDAG) is a DAG $G = (V, E)$ where we identify the vertices $V$ as

- **inputs**: $I \subseteq V$, the vertices with indegree-0,
- **outputs**: $O \subseteq V$, the vertices with outdegree-0, or
- **operations**: $Z = V \setminus I$ (the vertices with positive indegree);

we let $N = |Z|$, and suppose $I \cap O = \emptyset$.

**Definition**

A *schedule* $T$ of $G$ is a permutation $(v_1, \ldots, v_N)$ of $Z$ such that $v_i E v_j \Rightarrow i < j$, i.e., a topological ordering of $(Z, E \cap Z^2)$. 
A straight-line program schema is a sequence

\[ y_1 = f_1(x_{1,1}, \ldots, x_{1,n_1}), \]
\[ y_2 = f_2(x_{2,1}, \ldots, x_{2,n_2}), \]
\[ \vdots \]
\[ y_N = f_N(x_{N,1}, \ldots, x_{N,n_N}), \]

of operations involving variables \( V = \bigcup_{i=1}^{N} \{y_i\} \cup \{x_{i,1}, \ldots, x_{i,n_i}\} \) and function templates \( F = \bigcup_{i=1}^{N} f_i \), where each \( f_i \) is a placeholder for a partial function that depends on all its \( n_i \) arguments. We additionally suppose that \( n_1, \ldots, n_N \geq 1 \) and that each \( v \in V \) is assigned a value exactly once (single static assignment form).
There is a natural correspondence between CDAGs and straight-line program schemata; given such a schema, let

- \( Z = \{y_1, \ldots, y_N\} \) denote the results of the operations,
- \( I = V \setminus Z \), the \( x_{i,j} \) which only appear as input arguments,
- \( uE v \) iff \((\exists i \in \{1, \ldots, N\})y_i = v \land (\exists j \in \{1, \ldots, n_i\})x_{i,j} = u\),
- \( O = \) the subset of \( Z \) with no successors in \((V, E)\).

**Lemma**

Consider a straight-line program schema \( S \) and its induced CDAG \( G \). Every straight-line program schema that induces a CDAG \( G' \cong G \) can be obtained from \( S \) by:

- **Permuting the operations of \( S \) according to a schedule \( T \) of \( G \)**
- **Replacing each argument tuple \((x_{i,1}, \ldots, x_{i,n_i})\) with any surjective tuple over \( \{x_{i,1}, \ldots, x_{i,n_i}\} \)**
- **Replacing \( V \) by another set of variables \( W \)**
CDAG Execution = Sequence of (Memory) Configurations

**Definition**

Given a CDAG \( G = (V, E) \) and some \( e \notin V \), a (memory) configuration \( X \) is a sequence \((x_1, x_2, \ldots)\) over \( V \cup \{e\} \); we assume that \( x_i \neq e \) for finitely many \( i \).

- A *initial configuration* \( X_0 = I \cup \{e\} \)
- A *terminal configuration* \( X_N \supseteq O \)

**Definition**

An execution \((T, X)\) of a CDAG \( G = (V, E) \) consists of a schedule \( T = (v_1, \ldots, v_N) \) and a finite sequence \( X = (X_0, \ldots, X_N) \) of configurations (from initial to terminal) where for each \( i \in \{1, \ldots, N\} \),

- \( X_i \setminus X_{i-1} = \{v_i\} \), and
- \( \text{pred}(v_i) \subset X_{i-1} \).
Space and I/O Complexity (I/II)

Consider any execution \((T, X)\) of a given a CDAG \(G = (V, E)\). We will bound below certain costs associated to \((T, X)\) in terms of just \(T\), thus obtaining bounds that apply to a family of executions with the same order of operations, but differ in their assignments of variables to addresses (memory management).

**Definition**

Consider any schedule \(T\) of a CDAG \(G = (V, E)\). The width of \(T\) between operations \(i\) and \(i + 1\) (for \(i \in \{0, \ldots, N\}\)),

\[
\text{width}_T(i) = |\text{pred}(\{v_{i+1}, \ldots, v_N\}) \setminus \{v_{i+1}, \ldots, v_N\}| + |O \cap \{v_1, \ldots, v_i\}|,
\]

The space of \(T\), \(\text{space}_T = \max_{i \in \{0, \ldots, N\}}(\text{width}_T(i))\).

**Lemma**

For any schedule \(T\) of a CDAG \(G = (V, E)\), \(\text{space}_T \geq \max\{|I|, |O|\}\).
Lemma

Every execution \((T, X)\) of a CDAG \(G = (V, E)\) has a configuration \(X_i \in X\) with at least \(\text{space}_T\) nonempty cells (values \(v \neq e\)).

Unlike space, it is unclear how to define \(Q_M\) for an execution \((T, X)\): we need to first show how \((T, X)\) corresponds to a sequence \((a_1, \ldots, a_t)\) of memory accesses, in order to determine how many are to addresses \(\{M + 1, M + 2, \ldots\}\). Memory accesses are incurred when transitioning between configurations; for generality we will not constrain how these transitions are implemented (e.g., move vs. copy instructions), and instead exploit the following observation:

Lemma

For all executions with the same schedule \(T\), \(Q_M \geq \text{space}_T - M\).
Definition

Consider a schedule $T = (v_1, \ldots, v_N)$ of a CDAG $G = (V, E)$, and consider any nonempty consecutive subsequence $T' = (v_i, \ldots, v_j)$. The sub-CDAG (of $G$) induced by $T'$ with respect to $T$ is a sub-DAG $G' = (V', E')$ of $G$, where

$$Z' = \{v_i, \ldots, v_j\}, \quad I' = \text{pred}(Z') \setminus Z', \quad V' = I' \cup Z',$$

$$O' = Z' \cap (O \cup \text{pred}(\{v_{j+1}, \ldots, v_N\})), \quad E' = E \cap (Z' \times V');$$

We let $\text{space}_{T', T}$ denote $\text{space}_{T', (\text{w.r.t. } G')}$. 

Lemma

Consider any schedule $T$ of a CDAG $G = (V, E)$ and any $M \in \mathbb{N}$. Suppose there exists $K \in \{1, \ldots, N\}$ and $L \in \mathbb{N}$ such that we can write $T = T_1 \cdots T_K$, a concatenation of $K$ nonempty (consecutive) subsequences, where for each $j \in \{1, \ldots, K\}$, $\text{space}_{T_j, T} \geq M + L$. Then, $Q_M \geq KL$. 

**Theorem**

Given a CDAG $G = (V, E)$, suppose there exists a nondecreasing $F : \{1, \ldots, N\} \to [0, \infty)$ where, for all schedules $T$ of $G$ and all nonempty contiguous subsequences $T'$ of $T$, space $T', T \geq F(|T'|)$. Then,

$$Q_M \geq \max_{L \in \{1, \ldots, \lceil F(N) \rceil - M\}} L \left\lfloor \frac{N}{F^*(M+L)} \right\rfloor$$

for $M \in \{0, \ldots, \lceil F(N) \rceil - 1\}$ and $F^*(y) = \min\{x \in \mathbb{N} : \lceil F(x) \rceil \geq y\}$, and $Q_M \geq 0$ for $M \geq \lceil F(N) \rceil$.

**Corollary**

Suppose $F(x) = x^{1/\sigma}$ for some $\sigma \geq 1$. If $N \geq (M + 2)^\sigma$,

$$Q_M \geq M \left\lfloor \frac{N}{|M+1|^\sigma} \right\rfloor$$

If, additionally, $N > (M + 1)^{2\sigma}/((M + 1)^\sigma - 1) = \Omega(M^\sigma)$, then there exists a $c \in (0, \infty)$ such that $Q_M \geq cN/M^{\sigma-1}$; we express this as

$$Q_M = \Omega(N/M^{\sigma-1}).$$
An Isoperimetric Inequality (I/II)

Later, we will define a family of CDAGs sharing a certain geometric structure, and apply the following generalization of the discrete Loomis-Whitney inequality to derive a lower bound $F(|T'|)$ on space $T$, $T'$ for all nonempty subsequences $T'$ of all schedules $T$ of all $G$ in the family.

**Definition**

A *datum*, for some $d, m \in \mathbb{N}$ with $m \geq 1$, is a tuple $\phi = (\phi_1, \ldots, \phi_m)$ of group homomorphisms each with domain $\mathbb{Z}^d$.

**Theorem**

Consider a datum $\phi$ and $s \in [0, \infty)^m$.

\[
(\forall H \leq \mathbb{Z}^d) \quad \text{rank}(H) \leq \sum_{j=1}^{m} s_j \text{rank}(\phi_j(H))
\]

$\iff$

\[
(\forall \text{ finite nonempty } E \subseteq \mathbb{Z}^d) \quad |E| \leq \prod_{j=1}^{m} |\phi_j(E)|^{s_j}.
\]
Lemma

- The set of $s \in [0, \infty)^m$ that satisfy the previous inequalities define a closed, convex polytope $\mathcal{P} = \mathcal{P}(\phi)$, with finitely many faces.

- $\mathcal{P} \neq \emptyset$ iff $\bigcap_{j=1}^m \ker(\phi_j) = \{0\}$.

- For every $s \in \mathcal{P}$,
  - for any $[0, \infty)^m \ni t \geq s$ (componentwise), $t \in \mathcal{P}$;
  - there exists $t \in \mathcal{P} \cap [0, 1]^m$ with $t_j = \min\{1, s_j\}$ for $j \in \{1, \ldots, m\}$.

In particular, if $\mathcal{P} \neq \emptyset$ then $(1, \ldots, 1) \in \mathcal{P}$.

This lemma motivates us to restrict our attention to $\mathcal{P} \cap [0, 1]^m$. 

Definition

A CDAG $G = (V, E)$ has a geometric interpretation (or is geometric) if there exists $1 \leq d, m \in \mathbb{N}, K_1, \ldots, K_m \leq \mathbb{Z}^d$, and an injection $\psi : V \to \mathbb{Z}^d$ such that, for each $j \in \{1, \ldots, m\}$, for each $H \in \mathbb{Z}^d/K_j$, either $X = H \cap \psi(V) = \emptyset$ or $|X| \geq 2$ and

- $|X \cap (I \cup O)| \geq 1$,
- $|X \cap I| \leq 1$,
- $G|_X$ contains an arborescence on $X$ where each branch is length 1, except possibly for the trunk.

Every geometric CDAG corresponds to a family of data, each of the form $\phi = (\phi_1, \ldots, \phi_m)$, where, for each $j \in \{1, \ldots, m\}$, $\phi_j$ is any group homomorphism with domain $\mathbb{Z}^d$ and kernel $K_j$. Two geometric CDAGs with are similar if they have the same $d$ and $m$, and their $m$-tuples of subgroups $(K_j)$ and $(K'_j)$ are equal after embedding into $\mathbb{Q}^d$. 
Theorem

If a CDAG $G = (V, E)$ has a geometric interpretation with a datum $\phi$ where $\mathcal{P}(\phi) \neq \{0\}$, then for all $s \in \mathcal{P}(\phi)$, for all schedules $T$ of $G$, and for all nonempty consecutive subsequences $T'$ of $T$, space$_{T', T} \geq |T'|^{1/\sigma}$, where $\sigma = \sum_{j=1}^{m} s_j$.

- Note that this lower bound depends only on the datum $\phi$, not on $V$ or its embedding $\psi(V)$ into $\mathbb{Z}^d$, so it also applies to any DAG that is similar to $G$. Thus, we may apply the corollary above to conclude $Q_M = \Omega(N/M^{\sigma^{-1}})$ for any CDAG with datum $\phi$.
- The tightest lower bound given by this theorem picks $\sigma = \min_{s \in \mathcal{P}} \sum_{j=1}^{m} s_j$, a linear programming problem.
Example: Matrix Multiplication/Tensor Contraction

Suppose we have tensors $A$ and $B$ such that

- $A$ has $a + c$ modes of dimensions $m_1, \ldots, m_a, p_1, \ldots, p_c$,
- $B$ has $c + b$ modes of dimensions $p_1, \ldots, p_c, n_1, \ldots, n_b$,

and we wish to contract over the last $c$ modes of $A$, and the first $c$ modes of $B$,

$$C(i_1, \ldots, i_a, j_1, \ldots, j_b) = \sum_{k_1=1}^{p_1} \cdots \sum_{k_c=1}^{p_c} A(i_1, \ldots, i_a, k_1, \ldots, k_c) \cdot B(k_1, \ldots, k_c, j_1, \ldots, j_b).$$

We can linearize sets of modes and reindex, letting $m = \prod_{i=1}^{a} m_i$, $n = \prod_{i=1}^{b} n_i$, and $p = \prod_{i=1}^{c} p_i$, revealing

$$\hat{C}(i, j) = \sum_{k=1}^{p} \hat{A}(i, k) \cdot \hat{B}(k, j).$$

We obtain the communication lower bound

$$Q_M = \Omega(mp/n/M^{1/2}).$$
Example: Cartesian Products

Let $X_1, \ldots, X_d$ be finite sets of cardinalities $n_1, \ldots, n_d$; iterate over $X_1 \times \cdots \times X_d$, e.g.,

- Matrix/vector multiplication (implicit $A$)
  
  \[
  \text{for } i = 1 : m, \text{ for } j = 1 : n, \\
  y_i += A(i, j) \cdot x_j
  \]

- Direct $n$-body simulation (2-body interactions)
  
  \[
  \text{for } i = 1 : n, \text{ for } j = 1 : n, \\
  \text{force}_i += \text{interact}(p_i, p_j)
  \]

- Database joins, etc.

We obtain the communication lower bound

\[
Q_M = \Omega(n_1 \cdots n_d / M^{d-1}).
\]
Given $1 \leq d \in \mathbb{N}$, nonempty finite $Z \subset \mathbb{Z}^d$, and affine functions $\phi_1, \ldots, \phi_m: \mathbb{Z}^d \rightarrow \mathbb{Z}^d$; consider evaluating the recurrence equation

$$
 x(i) = f_i(x(\phi_1(i)), \ldots, x(\phi_m(i))),
$$

for each $i \in Z$.

- For example, uniform dependences $v_1, \ldots, v_m \in \mathbb{Z}^d$,

$$
 x(i) = f_i(x(i - v_1), \ldots, x(i - v_m)).
$$

(Finite difference methods, dynamic programming, . . . )

We obtain a communication lower bound of the form

$$
 Q_M = \Omega(|Z|/M^{\sigma-1}).
$$
Theorem

The polytope $\mathcal{P} = \mathcal{P}\left(\phi\right)$ is computable.

Approach 1

Embed $\mathbb{Z}^d$ into $\mathbb{R}^d$, reinterpret $\phi$ as real linear maps; for each $(r_0, \ldots, r_m) \in \{0, \ldots, d\}$, ask whether there exists a subspace $V \leq \mathbb{R}^d$ such that $\dim(V) = r_0$ and $\dim(\phi_j(V)) = r_j$ for $j \in \{1, \ldots, m\}$ (Tarski-decidable; cylindrical algebraic decomposition). If so, it induces a supporting hyperplane of $\mathcal{P}$.

Approach 2

Embed $\mathbb{Z}^d$ into $\mathbb{Q}^d$, reinterpret $\phi$ as rational linear maps; enumerate the subspaces $V_1, V_2, \ldots \in \mathbb{Q}^d$, generating $\mathcal{P}_1, \mathcal{P}_2, \ldots \supseteq \mathcal{P}$, stopping arbitrarily to ask whether $\mathcal{P}_i = \mathcal{P}$, by asking whether each extreme point $x \in \mathcal{P}_i$ is in $\mathcal{P}$ (decidable; compute $\mathcal{P}'(\phi')$ corresponding to $\mathbb{Q}^{d'}$ with $d' < d$ calling same procedure recursively).

There are many special cases where $\mathcal{P}$ can be computed more efficiently, e.g., $O(2^m \text{poly}(d))$ time.
Consider loop nests of the form

\[ y(\phi_0(i)) = f_i(y(\phi_0(i)), x_1(\phi_1(i)), \ldots, x_m(\phi_m(i))) \]

if \( \ker(\phi_0), \ldots, \ker(\phi_m) \) have bases \( K_0, \ldots, K_m \) over \( \mathbb{Z} \), and \( \bigcup_{j=0}^m K_j \subseteq K \), a basis of \( \mathbb{Z}^d \), then we can attain \( Q_M = O(n^d/M^{\sigma-1}) \) for sufficiently large \( n \) and \( M \) (and fixed \( d, m \)).

This includes the important case where all array references use subsets of the loop indices, e.g., \( A(i_1, i_2), B(i_3) \), as opposed to the general affine case, e.g., \( C(3i_1 - i_2 + 4) \).
Conclusions

- Many computations can be modeled as geometric CDAGs
- Derive communication lower bounds with isoperimetric inequalities
- Lower bounds attainable for important class of geometric CDAGs
- Ongoing work:
  - Efficiently computing lower bounds (algorithms for $\mathcal{P}$)
  - Attainability in general (affine) case (automatic block sizes?)
  - Polyhedral framework applications
  - Irregularly nested loops (by-statement scheduling)
  - Model blocked/pipelined communication (latency vs. bandwidth)

Thank You!
J. Bennett, A. Carbery, M. Christ, and T. Tao.
Finite bounds for Hölder-Brascamp-Lieb multilinear inequalities.

Minimizing communication in numerical linear algebra.

Communication lower bounds and optimal algorithms for programs that reference arrays — part I.

J.-W. Hong and H.T. Kung.
I/O complexity: the red-blue pebble game.

D. Irony, S. Toledo, and A. Tiskin.
Communication lower bounds for distributed-memory matrix multiplication.

L.H. Loomis and H. Whitney.
An inequality related to the isoperimetric inequality.