On Multi-Broadcast and Scheduling Receive-Graphs under LogP with Long Messages

Denis Trystram
IMAG, groupe ID
Grenoble, France

Wolf Zimmermann (speaker)
Institut für Informatik
Universität Halle
Halle, Germany
1. Motivation

- realistic abstract machine model for parallel MIMD computers with distributed memory
1. Motivation

- realistic abstract machine model for parallel MIMD computers with distributed memory

- optimizing parallel programs under this model
1. Motivation

- realistic abstract machine model for parallel MIMD computers with distributed memory
- optimizing parallel programs under this model
  - represent parallel programs as task graphs
1. Motivation

- realistic abstract machine model for parallel MIMD computers with distributed memory

- optimizing parallel programs under this model
  - represent parallel programs as task graphs
  - optimization problem $\cong$ scheduling problem
1. Motivation

- realistic abstract machine model for parallel MIMD computers with distributed memory

- optimizing parallel programs under this model
  - represent parallel programs as task graphs
  - optimization problem $\cong$ scheduling problem

How difficult is this optimization?
3 Related Results

4 Extended LogP-Model

5 Multibroadcast Problems

6 Scheduling Receive Graphs
2. Related Results

- task graphs under communication latencies:
2. Related Results

- task graphs under communication latencies:
  - general problem is NP-complete
2. Related Results

- task graphs under communication latencies:
  - general problem is NP-complete
  - trees of height one can be optimally scheduled to at most \( n \) processors in time \( O(n \log n) \)
2. Related Results

- **task graphs under communication latencies:**
  - general problem is NP-complete
  - trees of height one can be optimally scheduled to at most $n$ processors in time $O(n \log n)$

- **task graphs under LogP:**
  - NP complete to schedule trees of height one to at most $n$ processors (Verriet 1999)
2. Related Results

- task graphs under communication latencies:
  - general problem is NP-complete
  - trees of height one can be optimally scheduled to at most \( n \) processors in time \( O(n \log n) \)

- task graphs under LogP:
  - NP complete to schedule trees of height one to at most \( n \) processors (Verriet 1999)
  - approximation algorithms with performance ratio 2 (Verriet 1999)
2. Related Results

- task graphs under communication latencies:
  - general problem is NP-complete
  - trees of height one can be optimally scheduled to at most \( n \) processors in time \( O(n \log n) \)

- task graphs under LogP:
  - NP complete to schedule trees of height one to at most \( n \) processors (Verriet 1999)
  - approximation algorithms with performance ratio 2 (Verriet 1999)

Here: schedule trees of height one to at most \( P \) processors on LogP with long messages
3. The LogP Model
3. The LogP Model
3. The LogP Model

interconnection network

processors
3. The LogP Model

interconnection network

processors

local memory

local memory

local memory

local memory

local memory
3. The LogP Model
3. The LogP Model

Latency $L$:
maximal time between completion of send operation and start of receive operation
3. The LogP Model

Latency $L$:
maximal time between completion of send operation and start of receive operation

Overhead $\rho$:
time consumed by processor for sending/receiving a message
3. The LogP Model

Latency $L$:
maximal time between completion of send operation and start of receive operation

Overhead $o$:
time consumed by processor for sending/receiving a message

Gap $g$:
busy time of network connection of the sending/receiving processor
3. The LogP Model

**Latency** $L$
- maximal time between completion of send operation and start of receive operation

**Overhead** $\omega$
- time consumed by processor for sending/receiving a message

**Gap** $g$
- busy time of network connection of the sending/receiving processor

**Number of Processors** $P$
3. The LogP Model

Latency $L$
maximal time between completion of send operation and start of receive operation

Overhead $o$
time consumed by processor for sending/receiving a message

Gap $g$
busy time of network connection of the sending/receiving processor

Number of Processors $P$

Capacity Constraint:
- at any time: $\leq \left\lfloor L/g \right\rfloor$ messages in transit from/to any processor
3. The LogP Model

Latency \( L_0 = L_1 x \) message size \( x \)
maximal time between completion of send operation and start of receive operation

Overhead \( o_0 + o_1 x \)
time consumed by processor for sending/receiving a message

Gap \( g_0 + g_1 x \)
busy time of network connection of the sending/receiving processor

Number of Processors \( P \)
3. The LogP Model

Latency $L_0 = L_1 x$ message size $x$
maximal time between completion of send operation and start of receive operation

Overhead $o_0 + o_1 x$
time consumed by processor for sending/receiving a message

Gap $g_0 + g_1 x$
busy time of network connection of the sending/receiving processor

Number of Processors $P$

Capacity Constraint:
3. The LogP Model

Latency \( L_0 = L_1 x \) message size \( x \)
maximal time between completion of send operation and start of receive operation

Overhead \( o_0 + o_1 x \)
time consumed by processor for sending/receiving a message

Gap \( g_0 + g_1 x \)
busy time of network connection of the sending/receiving processor

Number of Processors \( P \)
Capacity Constraint: at any time
\[ \leq \left\lceil \frac{L_0}{g_0} \right\rceil \] messages in transit from/to any processor
3. The LogP Model

**Latency** \( L_0 = L_1 x \)  
message size \( x \)  
maximal time between completion of send operation and start of receive operation

**Overhead** \( o_0 + o_1 x \)  
time consumed by processor for sending/receiving a message

**Gap** \( g_0 + g_1 x \)  
busy time of network connection of the sending/receiving processor

**Number of Processors** \( P \)

**Capacity Constraint:** at any time
- \( \leq \left\lfloor L_0 / g_0 \right\rfloor \) messages in transit from/to any processor
- \( \leq \left\lfloor L_0 / g_1 \right\rfloor \) bytes in transit from/to any processor
# LogP Parameters for some Parallel Machines

<table>
<thead>
<tr>
<th>Machine</th>
<th>( L )</th>
<th>( o )</th>
<th>( g )</th>
<th>( P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM-5</td>
<td>(6 \mu s)</td>
<td>(2.2 \mu s)</td>
<td>(4 \mu s)</td>
<td>512</td>
</tr>
<tr>
<td>Parsytec Xplorer</td>
<td>(21 - 0.82x \mu s)</td>
<td>(70 + x \mu s)</td>
<td>(115 + 1.43x \mu s)</td>
<td>8</td>
</tr>
<tr>
<td>ParaStation</td>
<td>(50 - 0.1x \mu s)</td>
<td>(3 + 0.112x \mu s)</td>
<td>(3 + 0.119x \mu s)</td>
<td>4</td>
</tr>
<tr>
<td>IBM SP–1</td>
<td>1000 cycles</td>
<td>8000 cycles</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>IBM SP–2</td>
<td>(13 - 0.005x \mu s)</td>
<td>(8 + 0.008x \mu s)</td>
<td>(10 + 0.01x \mu s)</td>
<td>32</td>
</tr>
<tr>
<td>Meiko CS-2</td>
<td>(8.6 \mu s)</td>
<td>(1.7 \mu s)</td>
<td>(14.2 + 0.03x \mu s)</td>
<td>64</td>
</tr>
</tbody>
</table>
## LogP Parameters for some Parallel Machines

<table>
<thead>
<tr>
<th>Machine</th>
<th>L</th>
<th>o</th>
<th>g</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM-5</td>
<td>6 µs</td>
<td>2.2 µs</td>
<td>4 µs</td>
<td>512</td>
</tr>
<tr>
<td>Parsytec Xplorer</td>
<td>21 − 0.82x µs</td>
<td>70 + x µs</td>
<td>115 + 1.43x µs</td>
<td>8</td>
</tr>
<tr>
<td>ParaStation</td>
<td>50 − 0.1x µs</td>
<td>3 + 0.112x µs</td>
<td>3 + 0.119x µs</td>
<td>4</td>
</tr>
<tr>
<td>IBM SP–1</td>
<td>1000 cycles</td>
<td>8000 cycles</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>IBM SP–2</td>
<td>13 − 0.005x µs</td>
<td>8 + 0.008x µs</td>
<td>10 + 0.01x µs</td>
<td>32</td>
</tr>
<tr>
<td>Meiko CS-2</td>
<td>8.6 µs</td>
<td>1.7 µs</td>
<td>14.2 + 0.03x µs</td>
<td>64</td>
</tr>
</tbody>
</table>

- accurate performance predictions
- often: < 5% deviation between predicted and measured time
4. Multibroadcast Problems
4. Multibroadcast Problems

- all-to-one broadcast problem is dual
Multibroadcast under Communication Delays

\[ o = 0, \ g = 0 \]
\( o = 0, g = 0 \)

\[ \pi \] sends \( M_1, \ldots, M_i \) at time 0
Multibroadcast under Communication Delays

\( o = 0, g = 0 \)

- \( \pi \) sends \( M_1, \ldots, M_i \) at time 0

\( \implies \) optimal
Multibroadcast under Classical LogP
Multibroadcast under Classical LogP

- \( \pi \) sends \( M_1, \ldots, M_i \) sequentially
Multibroadcast under Classical LogP

- \( \pi \) sends \( M_1, \ldots, M_i \) sequentially

\[ L = 2, \; o = 10, \; g = 0, \; P = 9 \]
Multibroadcast under Classical LogP

- $\pi$ sends $M_1, \ldots, M_i$ sequentially

$L = 2, o = 10, g = 0, P = 9$
Multibroadcast under Classical LogP

- \( \pi \) sends \( M_1, \ldots, M_i \) sequentially

\[ L = 2, \ o = 10, \ g = 0, \ P = 9 \]

\[ \Rightarrow \text{optimal} \]
The General Case

Idea:  merge messages to large ones
The General Case

**Idea:** merge messages to large ones

**Multi-Broadcast Tree:** represent multi-broadcast algorithms
The General Case

**Idea:** merge messages to large ones

**Multi-Broadcast Tree:** represent multi-broadcast algorithms

\[ L = 2 - x, \quad o = 10 + x, \quad g = 0, \quad P = 9 \]
The General Case

Idea: merge messages to large ones

Multi-Broadcast Tree: represent multi-broadcast algorithms

\[ L = 2 - x, \ o = 10 + x, \ g = 0, \ P = 9 \]
The General Case

Idea: merge messages to large ones

Multi-Broadcast Tree: represent multi-broadcast algorithms

\[ L = 2 - x, \quad o = 10 + x, \quad g = 0, \quad P = 9 \]

- Label \( t \) at node \( l \): time when processor \( \pi_l \) has its message available
The General Case

**Idea:** merge messages to large ones

**Multi-Broadcast Tree:** represent multi-broadcast algorithms

\[ L = 2 - x, \ o = 10 + x, \ g = 0, \ P = 9 \]

Label \( t \) at node \( l \): time when processor \( \pi_l \) has its message available
The General Case

**Idea:** merge messages to large ones

**Multi-Broadcast Tree:** represent multi-broadcast algorithms

\[ L = 2 - x, \quad o = 10 + x, \quad g = 0, \quad P = 9 \]

- Label \( t \) at node \( l \): time when processor \( \pi_l \) has its message available
- labels can be computed, details see paper.
The General Case

Idea: merge messages to large ones

Multi-Broadcast Tree: represent multi-broadcast algorithms

$L = 2 - x$, $o = 10 + x$, $g = 0$, $P = 9$

- Label $t$ at node $l$: time when processor $\pi_l$ has its message available
- labels can be computed, details see paper.
The General Case

**Idea:** merge messages to large ones

**Multi-Broadcast Tree:** represent multi-broadcast algorithms

\[ L = 2 - x, \quad o = 10 + x, \quad g = 0, \quad P = 9 \]

- Label \( t \) at node \( l \): time when processor \( \pi_l \) has its message available
- labels can be computed, details see paper.
The General Case

Idea: merge messages to large ones

Multi-Broadcast Tree: represent multi-broadcast algorithms

\[ L = 2 - x, \quad o = 10 + x, \quad g = 0, \quad P = 9 \]

- Label \( t \) at node \( l \): time when processor \( \pi_l \) has its message available
- labels can be computed, details see paper.
The General Case

**Idea:** merge messages to large ones

**Multi-Broadcast Tree:** represent multi-broadcast algorithms

$L = 2 - x$, $o = 10 + x$, $g = 0$, $P = 9$

- Label $t$ at node $l$: time when processor $\pi_l$ has its message available
- labels can be computed, details see paper.
The General Case

Idea: merge messages to large ones

Multi-Broadcast Tree: represent multi-broadcast algorithms

$L = 2 - x, o = 10 + x, g = 0, P = 9$

Label $t$ at node $l$: time when processor $\pi_l$ has its message available

- labels can be computed, details see paper.
The General Case

**Idea:** merge messages to large ones

**Multi-Broadcast Tree:** represent multi-broadcast algorithms

\[ L = 2 - x, \quad o = 10 + x, \quad g = 0, \quad P = 9 \]

- Label \( t \) at node \( l \): time when processor \( \pi_l \) has its message available
- labels can be computed, details see paper.
The General Case

Optimal Multi-Broadcast Tree:

largest label \leq \text{largest label for any multi-broadcast tree}
The General Case

Optimal Multi-Broadcast Tree:

largest label \(\leq\) largest label for any multi-broadcast tree

**Lemma** There is an optimal multi-broadcast tree such that the size of the subtrees of the root can be ordered non-increasingly

**Construction:**

1. determine multi-broadcast tree with \(k\) subtrees that is optimal among these
**The General Case**

**Optimal Multi-Broadcast Tree:**

largest label \( \leq \) largest label for any multi-broadcast tree

**Lemma** There is an optimal multi-broadcast tree such that the size of the subtrees of the root can be ordered non-increasingly

**Construction:**

1. determine multi-broadcast tree with \( k \) subtrees that is optimal among these
   1. determine recursively optimal multi-broadcast trees for \( j_i \) processors, \( i = 1, \ldots, k \)
The General Case

**Optimal Multi-Broadcast Tree:**

largest label $\leq$ largest label for any multi-broadcast tree

**Lemma** There is an optimal multi-broadcast tree such that the size of the subtrees of the root can be ordered non-increasingly

**Construction:**

1. determine multi-broadcast tree with $k$ subtrees that is optimal among these
   - determine recursively optimal multi-broadcast trees for $j_i$ processors, $i = 1, \ldots, k$
   - construct multibroadcast tree with these subtrees
**The General Case**

**Optimal Multi-Broadcast Tree:**

largest label \( \leq \) largest label for any multi-broadcast tree

**Lemma** There is an optimal multi-broadcast tree such that the size of the subtrees of the root can be ordered non-increasingly

**Construction:**

1. determine multi-broadcast tree with \( k \) subtrees that is optimal among these
   - determine recursively optimal multi-broadcast trees for \( j_i \) processors, \( i = 1, \ldots, k \)
   - construct multibroad-cast tree with these subtrees
   - \( T'(j_1, \ldots, j_k) \) largest label

\[
T''(k) = \min_{j_1 + \cdots + j_k = P-1} T'(j_1, \ldots, j_k)
\]

\[j_1 \geq \cdots \geq j_k \geq 1\]
The General Case

**Optimal Multi-Broadcast Tree:**

largest label ≤ largest label for any multi-broadcast tree

**Lemma** There is an optimal multi-broadcast tree such that the size of the subtrees of the root can be ordered non-increasingly

**Construction:**

1. determine multi-broadcast tree with $k$ subtrees that is optimal among these
   - determine recursively optimal multi-broadcast trees for $j_i$ processors, $i = 1, \ldots, k$
   - construct multibroad-cast tree with these subtrees
   - $T'(j_1, \ldots, j_k)$ largest label

   $$T''(k) = \min_{j_1 + \cdots + j_k = P-1 \atop j_1 \geq \cdots \geq j_k \geq 1} T'(j_1, \ldots, j_k)$$

2. The optimal multi-broadcast tree is one of those such that $T''(k)$ is minimal for $k = 1, \ldots, P$

**Execution Time:** exponential
5. Scheduling Receive Graphs

**Theorem:**

- $P \geq 2$ processors
- extended LogP model with $o_1 > 0$, $g_0 = o_0$, $g_1 = o_1$
- integer $B$
- receive graph for $P$ processors

NP-complete to decide whether there is a schedule with makespan at most $B$
5. Scheduling Receive Graphs

**Theorem:**

- $P \geq 2$ processors
- extended LogP model with $o_1 > 0, g_0 = o_0, g_1 = o_1$
- integer $B$
- receive graph for $P$ processors

NP-complete to decide whether there is a schedule with makespan at most $B$

**Corollary:** also NP-complete for the multi-broadcast problem
Reduction from \textsc{Partition}:

\textbf{Input:} A set \( A = \{ a_1, \ldots, a_n \} \subseteq \mathbb{N}, \ a_i > 0. \)

\textbf{Question:} Is there a subset \( A' \subseteq A \) such that \( \sum_{a \in A'} a = \frac{1}{2} \cdot \sum_{a \in A} a? \)

\[
\begin{align*}
\tau(r) &= 1 \\
c &= (2o_1 - L_1) \cdot (L_0 + o_0 + 1) \cdot S \\
c' &= c \cdot P \\
c'' &= (c + c') \cdot S - 2o_1 + L_1 \\
\tau(l_i) &= 2 \cdot c' \cdot a_i \\
\tau(m_j) &= c'' + j \cdot o_0 \\
\sigma(l_i) &= 2 \cdot (L_0 + o_0 + 1) a_i \\
\sigma(m_j) &= 1 \\
B &= c'' + L_0 + o_0 \cdot P + 1.
\end{align*}
\]
Reduction from \textsc{Partition}:

\textbf{Input:} A set \( A = \{a_1, \ldots, a_n\} \subseteq \mathbb{N}, \ a_i > 0. \)

\textbf{Question:} Is there a subset \( A' \subseteq A \) such that
\[ \sum_{a \in A'} a = \frac{1}{2} \cdot \sum_{a \in A} a? \]

Suppose \( A_1 \) is solution of partition

\[
\begin{align*}
\tau(r) &= 1 \\
c &= (2o_1 - L_1) \cdot (L_0 + o_0 + 1) \cdot S \\
c' &= c \cdot P \\
c'' &= (c + c') \cdot S - 2o_1 + L_1 \\
\tau(l_i) &= 2 \cdot c' \cdot a_i \\
\tau(m_j) &= c'' + j \cdot o_0 \\
\sigma(l_i) &= 2 \cdot (L_0 + o_0 + 1) a_i \\
\sigma(m_j) &= 1 \\
B &= c'' + L_0 + o_0 \cdot P + 1.
\end{align*}
\]
**Reduction from Partition:**

**Input:** A set $A = \{a_1, \ldots, a_n\} \subseteq \mathbb{N}$, $a_i > 0$.

**Question:** Is there a subset $A' \subseteq A$ such that $\sum_{a \in A'} = \frac{1}{2} \cdot \sum_{a \in A}$?

Suppose $A_1$ is solution of partition

\[
\begin{align*}
\tau(r) &= 1 \\
c &= (2o_1 - L_1) \cdot (L_0 + o_0 + 1) \cdot S \\
c' &= c \cdot P \\
c'' &= (c + c') \cdot S - 2o_1 + L_1 \\
\tau(l_i) &= 2 \cdot c' \cdot a_i \\
\sigma(l_i) &= 2 \cdot (L_0 + o_0 + 1) a_i \\
\tau(m_j) &= c'' + j \cdot o_0 \\
\sigma(m_j) &= 1 \\
B &= c'' + L_0 + o_0 \cdot P + 1.
\end{align*}
\]
Reduction from **Partition:**

**Input:** A set $A = \{a_1, \ldots, a_n\} \subseteq \mathbb{N}$, $a_i > 0$.

**Question:** Is there a subset $A' \subseteq A$ such that $\sum_{a \in A'} = \frac{1}{2} \cdot \sum_{a \in A} a$?

Suppose $A_1$ is solution of partition

- $\tau(r) = 1$
- $c = (2o_1 - L_1) \cdot (L_0 + o_0 + 1) \cdot S$
- $c' = c \cdot P$
- $c'' = (c + c') \cdot S - 2o_1 + L_1$
- $\tau(l_i) = 2 \cdot c' \cdot a_i$
- $\tau(m_j) = c'' + j \cdot o_0$
- $\sigma(l_i) = 2 \cdot (L_0 + o_0 + 1)a_i$
- $\sigma(m_j) = 1$

$B = c'' + L_0 + o_0 \cdot P + 1$.

- if there is a schedule with makespan $\leq B$, it has this form
  
  details see paper
6. Conclusions

- realistic machine model: LogP extended with long messages
6. Conclusions

- realistic machine model: LogP extended with long messages
- multi-broadcast problem:
6. Conclusions

- realistic machine model: LogP extended with long messages
- multi-broadcast problem:
  - optimal solution in linear time:
6. Conclusions

- realistic machine model: LogP extended with long messages

- multi-broadcast problem:
  - optimal solution in linear time:
    - communication delay model
6. Conclusions

- realistic machine model: LogP extended with long messages
- multi-broadcast problem:
  - optimal solution in linear time:
    - communication delay model
    - classical LogP (with small messages)
6. Conclusions

- realistic machine model: LogP extended with long messages

- multi-broadcast problem:
  - optimal solution in linear time:
    - communication delay model
    - classical LogP (with small messages)
  - NP-complete for LogP with extended messages
6. Conclusions

- realistic machine model: LogP extended with long messages

- multi-broadcast problem:
  - optimal solution in linear time:
    - communication delay model
    - classical LogP (with small messages)
  - NP-complete for LogP with extended messages

⇒ heuristics for optimizing parallel programs
6. Conclusions

- realistic machine model: LogP extended with long messages

- multi-broadcast problem:
  - optimal solution in linear time:
    - communication delay model
    - classical LogP (with small messages)
  - NP-complete for LogP with extended messages

⇒ heuristics for optimizing parallel programs

Remark: single-broadcast problem and all-to-one broadcast on classical LogP is also NP-complete (to appear in IPL)