Distributed Execution

William Cook and Jayadev Misra

Department of Computer Science
University of Texas at Austin

Email: {cook, misra}@cs.utexas.edu
web: http://www.cs.utexas.edu/users/psp
Centralized Execution Model

- semantics (and execution) described for one machine.

- Then,
  - immediate sites respond instantaneously.
  - Rtimer is exact.

- We develop a theory to permit distributed execution.
Distributed Execution Model

- Assign subexpressions to different machines.
- Execution starts on a goal machine.
- A machine requests another machine to start evaluation.
- The calling machine supplies the context and values of variables as they become defined.
- The calling machine orders all tokens to be killed, and asks for ack.
- All messages among these machines are delivered after arbitrary but finite delay.
When can we distribute?

- $Rtimer(1) \gg let(0) \mid Rtimer(2) \gg let(1)$

Neither branch can be executed on another machine.

- $P(c, e) \Delta c.get \triangleright x \triangleright Compute(x) \triangleright y \triangleright e.put(y) \gg P(c, e)$
- $N \Delta P(c, e) \mid P(d, e)$

$c.get$ and $e.put$ are on different machines.
$P(c, e)$ and $P(d, e)$ can be evaluated on different machines.

- $Rtimer(1) \gg M \mid Rtimer(2) \gg N$

The two branches can be executed on different machines.

The theory identifies subexpressions which can be distributed.
Distributed Execution has to be faithful

- A distributed execution is valid if it mimics a centralized execution.
- The sites can not distinguish between the two execution styles.
An example

- Site *keyboard* responds at arbitrary times, or never. Its response is received immediately.

- Site *screen* responds immediately and displays a message.

  \[\cdots \text{keyboard} \Rightarrow \text{screen} \cdots\]

  can not be distributed.

**Video-game:**
A human responds sometime on the keyboard.

Expect to see an echo immediately.
A site is **punctual** if communication delay with it is zero.

A site is **unpunctual** if communication delay with it is arbitrary.

**Simplification**: Assume all sites are either punctual or unpunctual.

A punctual site has to be implemented on the caller’s machine.

An unpunctual site may be implemented on a remote machine.

**Punctual**: `Rtimer`, `let`, `if`, ···, all immediate sites, ···, `keyboard`, `screen`

**Unpunctual**: `0`, `CNN`, `PostWeb`, ···  Generic symbol `M`
Punctual Expression

- Any delay in starting it or processing its response can be detected.
- Some timing information for site calls or publications is known.
- A punctual expression has to be implemented on the caller’s machine.
- An unpunctual expression may be implemented on another machine.
- A punctual expression may have unpunctual subexpressions.
  The subexpression may be implemented on another machine.
Definition of Punctual Expression

A punctual expression is of the form

- \( S(x) \): \( S \) is punctual,
- \( f \mid g \): either \( f \) or \( g \) is punctual,
- \( f > x > g \): both \( f \) and \( g \) are punctual,
- \( f \text{ where } x \in g \): both \( f \) and \( g \) are punctual, or \( f \backslash x \) is punctual.

Obtain \( f \backslash x \) from \( f \):
Replace all site calls which have parameter \( x \) by \( 0 \).
Examples: Punctual Expressions

\[ \text{let}(x) \]  
\[ \text{let}(x) \triangleright x \triangleright \text{Rtimer}(x) \]  
\[ \text{let} \text{ is punctual (immediate)} \]
\[ \text{both sites are punctual} \]

\[ \text{Rtimer}(1) \text{ where } x : \in g \]  
\[ f \backslash x = \text{Rtimer}(1) \text{ is punctual} \]

\[ \text{let}(x) \text{ where } x : \in \text{Rtimer}(1) \]  
\[ \text{let} (x) \text{ and } \text{Rtimer}(1) \text{ are both punctual} \]

\[ \text{Rtimer}(1) \mid h \text{ where } x : \in g \]  
\[ f \backslash x = \text{Rtimer}(1) \mid h \backslash x \text{ is punctual} \]

\[ \text{let}(x) \triangleright x \triangleright \text{Rtimer}(1) \]  
\[ \text{where } x : \in \text{Rtimer}(1) \]  
\[ \text{let}(x) \triangleright x \triangleright \text{Rtimer}(1) \text{ and } \text{Rtimer}(1) \]  
\[ \text{are both punctual} \]
Examples: Unpunctual Expressions

\[ M \gg Rtimer(1) \]

\[ Rtimer(1) \gg M \]

\[ N(x) > x > Rtimer(x) \]

\[ Rtimer(x) \text{ where } x \in M \]

\[ N(x) \text{ where } x \in Rtimer(1) \]

\[ let(x) \text{ where } x \in M \]
punctual within unpunctual, and vice versa

\[ \text{keyboard} \rightarrow \text{screen} \mid \text{JoyStick} \rightarrow (M \mid N), \text{ where} \]

\[ \text{keyboard, screen, JoyStick are punctual} \]
\[ M \text{ and } N \text{ are unpunctual.} \]
**Example: Distributed Execution**

\[
\text{keyboard} \rightarrow \text{screen} \quad | \quad \text{JoyStick} \rightarrow (M \mid N)
\]

\[
\equiv \quad \text{keyboard} \rightarrow \text{screen} \mid f
\quad f \triangleleft \text{JoyStick} \rightarrow (M \mid N)
\]

\[
\equiv \quad \text{keyboard} \rightarrow \text{screen} \mid f
\quad f \triangleleft \text{JoyStick} \rightarrow g
\quad g \triangleleft M \mid N
\]
Some properties

- If all sites in an expression are punctual, the expression is punctual.
- If all sites in an expression are unpunctual, the expression is unpunctual.
- **Monotonicity**: An unpunctual expression remains unpunctual if you replace any site by an unpunctual site.

\[ M \text{ punctual, } f(M) \text{ unpunctual, } N \text{ unpunctual} \]
\[ \Rightarrow f(N) \text{ unpunctual.} \]

\[ M \sqsubseteq N \Rightarrow f(M) \sqsubseteq f(N), \]
where punctual \(\sqsubseteq\) unpunctual

Proofs by structural induction.
Punctuality is conservative

- \( \text{let(} \text{true} \text{)} \)
  \[
  \begin{align*}
  >b> & \quad ( \text{if}(b) \implies g) \quad \equiv \quad g \\
  & \quad | \text{if}(\neg b) \implies \text{Rtimer}(1) \\
  & \quad )
  \end{align*}
  \]
  Though the execution can be distributed, the expression is punctual.

- \( \text{Random} \ >x> \text{Rtimer}(x) \)

  \( \text{Random} \) publishes a random natural number within some bounds. The expression is punctual, and it can not be distributed.

- \( \text{Nat} \ >x> \text{Rtimer}(x) \)

  \( \text{Nat} \) publishes any natural number. Though the execution can be distributed, the expression is punctual.
Recursively defined Expressions

\[ \text{Metronome} \triangleq \text{Signal} \mid \text{Rtimer}(1) \Rightarrow \text{Metronome}, \text{ i.e.} \]
\[ \text{Metronome} \triangleq \text{if (true)} \mid \text{Rtimer}(1) \Rightarrow \text{Metronome} \]

\text{Signal} \text{ is punctual, so the rhs is punctual. } \text{Metronome} \text{ is punctual.} \]

\[ \text{Met} \triangleq \text{Signal} \mid \underline{\text{M} \Rightarrow \text{Met}} \]

\text{unpunctual} \]

\text{Met} \text{ is punctual. Its subexpression } \text{M} \Rightarrow \text{Met} \text{ may be distributed.} \]

\[ \text{Signal} \mid \text{M} \Rightarrow \text{Met} \]

\[ \equiv \quad \text{Signal} \mid f \]
\[ f \triangleq \text{M} \Rightarrow \text{Met} \]
The least fixed point is $f$ is punctual. punctual $\sqsubseteq$ unpunctual.
Existence of least fixed point

Least fixed point exists for any defined expression $f$.

Assume $f$ is punctual.

- definition of $f$ is punctual: $f$ is punctual.

- definition of $f$ is unpunctual: $f$ is unpunctual.

**Monotonicity**: assuming $f$ unpunctual, definition of $f$ is unpunctual.