Demo of ISP
Eclipse GUI
Command-line Options
Set-up Audience with LiveDVD

About 30 minutes - by Ganesh
Understanding Out-of-order Execution

- For MPI programs, no dynamic tool can maintain in-order (in program order) issue of all instructions and yet cover the full extent of non-determinism!

- Luckily, the MPI semantics are such that out-of-order issue of ‘hijacked’ instructions is OK!

- Theory of ‘happens-before’ (after break) justifies

- Illustration on Crooked-barrier now
  - Real code illustration is through POE-Illustration
Demonstrate on Auto-send Examples

- **Experiment with Autosend examples**
  - MPI_Autosend1 and Autosend2
    - These execute without deadlocks
    - Stepping in internal issue order shows how ISP performs things
    - The IntraCompletesBefore (recently called IntraHappensBefore) justifies this out-of-order execution
  - MPI_Autosend3
    - This example deadlocks
    - Pattern is $R$($from:0,h1$); $W(h1)$; $B$; $S$($to:0,h2$); $W(h2)$
  - Try variant MPI_Autosend4
    - Have pattern $S$($to:0,h2$); $W(h2)$; $B$; $R$($from:0,h1$); $W(h1)$
    - Run with and without buffering
    - What do you observe?
      - The case with buffering does not deadlock!
### ‘Crooked Barrier’ in POE-Illustration.c

<table>
<thead>
<tr>
<th>Process P0</th>
<th>Process P1</th>
<th>Process P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isend(1, req) ;</td>
<td>Irecv(*, req) ;</td>
<td><strong>Barrier</strong> ;</td>
</tr>
<tr>
<td><strong>Barrier</strong> ;</td>
<td><strong>Barrier</strong> ;</td>
<td>Isend(1, req) ;</td>
</tr>
<tr>
<td>Wait(req) ;</td>
<td>Recv(2) ;</td>
<td>Wait(req) ;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The “Crooked Barrier” example

Can $S_2$ (to : P2) match R(from : *)?
The “Crooked Barrier” example

Can \( S_2 \text{ (to : P2) } \) match \( R\text{(from : *)} \)?  YES! Here is how!
The “Crooked Barrier” example

Can $S_2 (\text{to : P2})$ match $R(\text{from : *})$? YES! Here is how!
The “Crooked Barrier” example

Can \( S_2 \) (to : P2) match \( R \) (from : *)? YES! Here is how!
The “Crooked Barrier” example

Can $S_2 \text{(to : P2)}$ match $R\text{(from : *)}$? YES! Here is how!
The “Crooked Barrier” example

Can $S_2 \text{ (to : P2) }$ match $R\text{ (from : *) }$?  YES!  Here is how!
The “Crooked Barrier” example

Can \( S_2 \text{ (to : P2) } \) match \( R \text{ (from : *) } \)? YES! Here is how!
The “Crooked Barrier” example

Can $S_2$ (to : P2 ) match $R$ (from : * )?  YES! Here is how!
The “Crooked Barrier” example

ISP handles this situation through
• Out-of-order execution
• Dynamic Instruction Rewriting
The “Crooked Barrier” example

ISP handles this situation through

• Out-of-order execution
• Dynamic Instruction Rewriting
The “Crooked Barrier” example

ISP handles this situation through
• Out-of-order execution
• Dynamic Instruction Rewriting
The “Crooked Barrier” example

P0
---
Collect
$S_1 \ (to : P2 );$

P1
---
B;

P2
---
Collect
$R(from : *);$

$B$
---
Issue into MPI runtime

$B$
---
Issue into MPI runtime

$B$
---
Issue into MPI runtime
The “Crooked Barrier” example

P0
---
Collect
\[ S_1 \ ( \text{to} : \ P2 \ ) ; \]
B

P1
---
B;

P2
---
Collect
\[ R( \text{from} : * ) ; \]

\[ S_2 \ ( \text{to} : \ P2 \ ) \]
B
The “Crooked Barrier” example

P0
---
Collect
\( S_1 \ (to : P2 ) \);  
B

P1
---
B;
Collect
\( S_2 \ (to : P2 ) \)

P2
---
Collect
\( R( from : * ) \);
B
The “Crooked Barrier” example

$S_1$ (to : P2 );

B;

$S_2$ (to : P2 )

B

R(from : *);

Form Matches
The “Crooked Barrier” example

P0 ---

$S_1$ (to : P2);

B

---

P1

B;

---

P2

$S_2$ (to : P2)

---

R(from : 0);

B

Rewrite, play
The “Crooked Barrier” example

P0
---
$S_1 \,(to : P_2 )$;
B

P1
---
B;

P2
---

R(from : *);
$S_2 \,(to : P_2 )$
B

Re-execute to get back here
The “Crooked Barrier” example

```
P0
---
Collect
S₁ (to : P2 );
B

P1
---
B;
Collect
S₂ (to : P2 )

P2
---
Collect
R(from : 1);
B
```

 Rewrite, play
Hijack Calls, Generate Relevant Interleavings

P0

- Isend(1, req)
- Barrier
- Wait(req)

P1

- Irecv(*, req)
- Barrier
- Recv(2)
- Wait(req)

P2

- Isend(1, req)
- Barrier
- Wait(req)

SendNext

Scheduler

- Isend(1)
- Barrier

MPI Runtime

23
Hijack Calls, Generate Relevant Interleavings

Isend(1, req)
Barrier
Wait(req)

Irecv(*, req)
Barrier

Irecv(*)
Barrier

Isend(1)
sendNext
Barrier

Recv(2)
Wait(req)

Wait(req)

MPI Runtime

Scheduler

Isend(1)
Barrier

Irecv(*)
Barrier
Hijack Calls, Generate Relevant Interleavings

MPI RunTime

P0
- Isend(1, req)
- Barrier
- Wait(req)

P1
- Irecv(*, req)
- Barrier
- Recv(2)
- Wait(req)

P2
- Isend(1, req)
- Barrier
- Irecv(*)
- Barrier
- Barrier
- Barrier
- Barrier

Scheduler
- Isend(1)
- Barrier
- Irecv(*)
- Barrier
- Barrier
Hijack Calls, Generate Relevant Interleavings

Isend(1, req)

Irecv(*, req)

Irecv(2)

Irecv(*)

Irecv(2)

Irecv

SendNext

No Match-Set

Barrier

Wait (req)

Barriere

Wait (req)

Barrier

Wait (req)

SendNext

Wait (req)

Isend

Isend(1)

Isend(1)

Deadlock!

Hijack Calls, Generate Relevant Interleavings

MPI Runtime

Scheduler

Isend(1)

Irecv(2)

Irecv

SendNext

No Match-Set

Barrier

Wait (req)

Isend(1)

Isend(1)

Deadlock!

Hijack Calls, Generate Relevant Interleavings

MPI Runtime

Scheduler

Isend(1)

Irecv(2)

Irecv

SendNext

No Match-Set

Barrier

Wait (req)

Isend(1)

Isend(1)

Deadlock!

Hijack Calls, Generate Relevant Interleavings

MPI Runtime

Scheduler

Isend(1)

Irecv(2)

Irecv

SendNext

No Match-Set

Barrier

Wait (req)

Isend(1)

Isend(1)

Deadlock!

Hijack Calls, Generate Relevant Interleavings

MPI Runtime

Scheduler

Isend(1)

Irecv(2)

Irecv

SendNext

No Match-Set

Barrier

Wait (req)

Isend(1)

Isend(1)

Deadlock!
Example 1: Any-src-can-deadlock9.c (Umpire)
lucky.c and unlucky.c are just variations

- **Distribute LiveDVDs**
  - Help audience boot into it

- **Source code of any-src-can-deadlock.c**
  - Demonstrate Verification Options
  - Observe completes-before

- **Source code of POE-illustration.c**
  - Observe completes-before
  - Observe internal issue order and program order
LiveDVD Instructions

• For Windows, boot using LiveDVD
  – Select ‘try Linux without installing

• For MAC, select boot device with LiveDVD present inside drive
  – Select to be DVD
  – Power down
  – Power-up with DVD in drive

• For MAC, you can restore boot device by pressing and holding down ALT / Option when booting again
Observations of examples lucky / unlucky

• ISP finds deadlock

• In Java GUI, do these
  – Watch the execution deadlock
  – Learn how to step
    • Step into different interleavings
    • Step through the traces of an interleaving
  – Do the above with and without rank locking
  – Do the above by locking ranks and watching
    • According to Program Order
    • According to Internal Issue Order

• Watch the IntraCB and InterCB
End of C