

# A New Approach to Parallel Debugger Architecture

Susanne M. Balle

Joint work with:

Bevin R. Brett

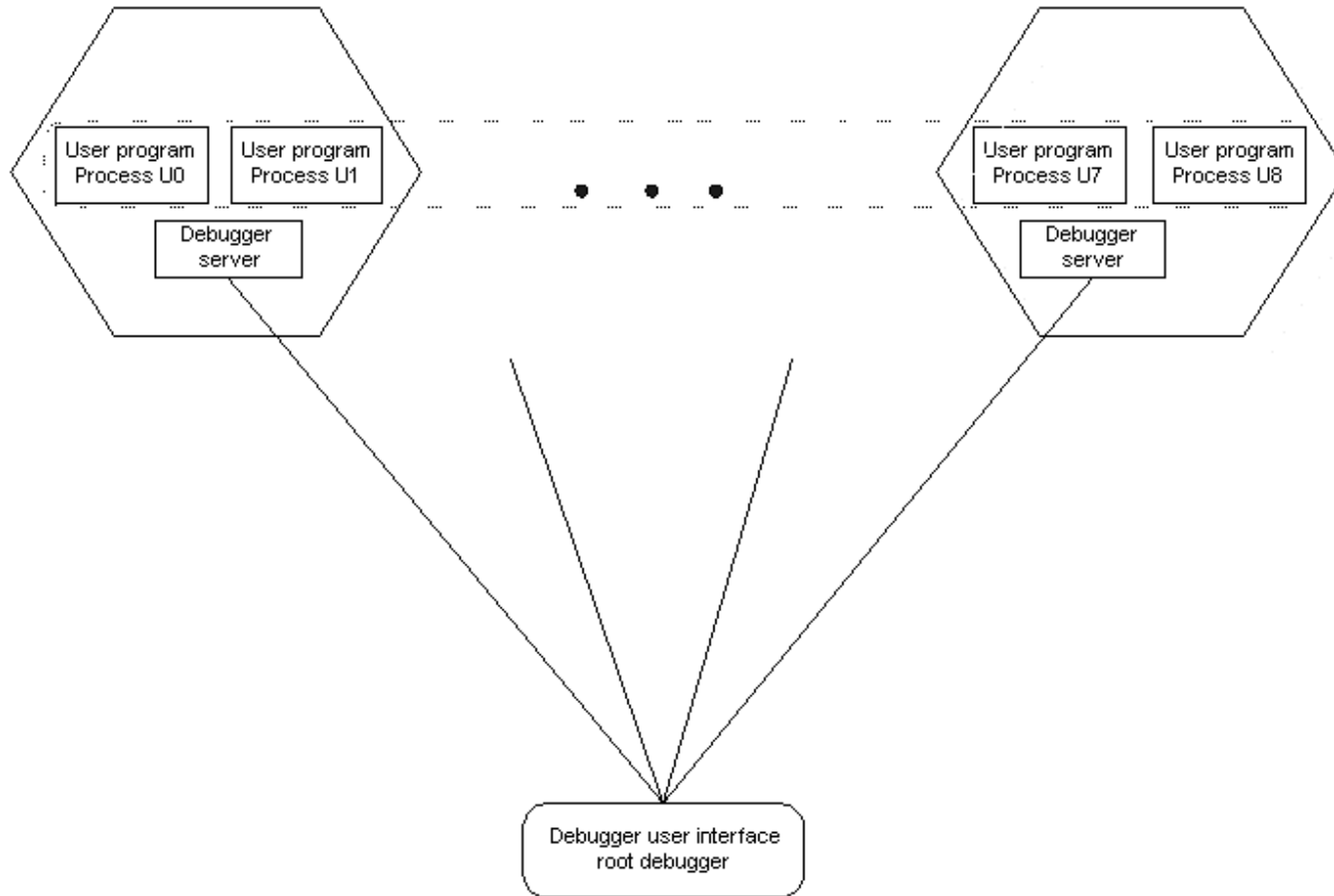
Chih-Ping Chen

David LaFrance-Linden

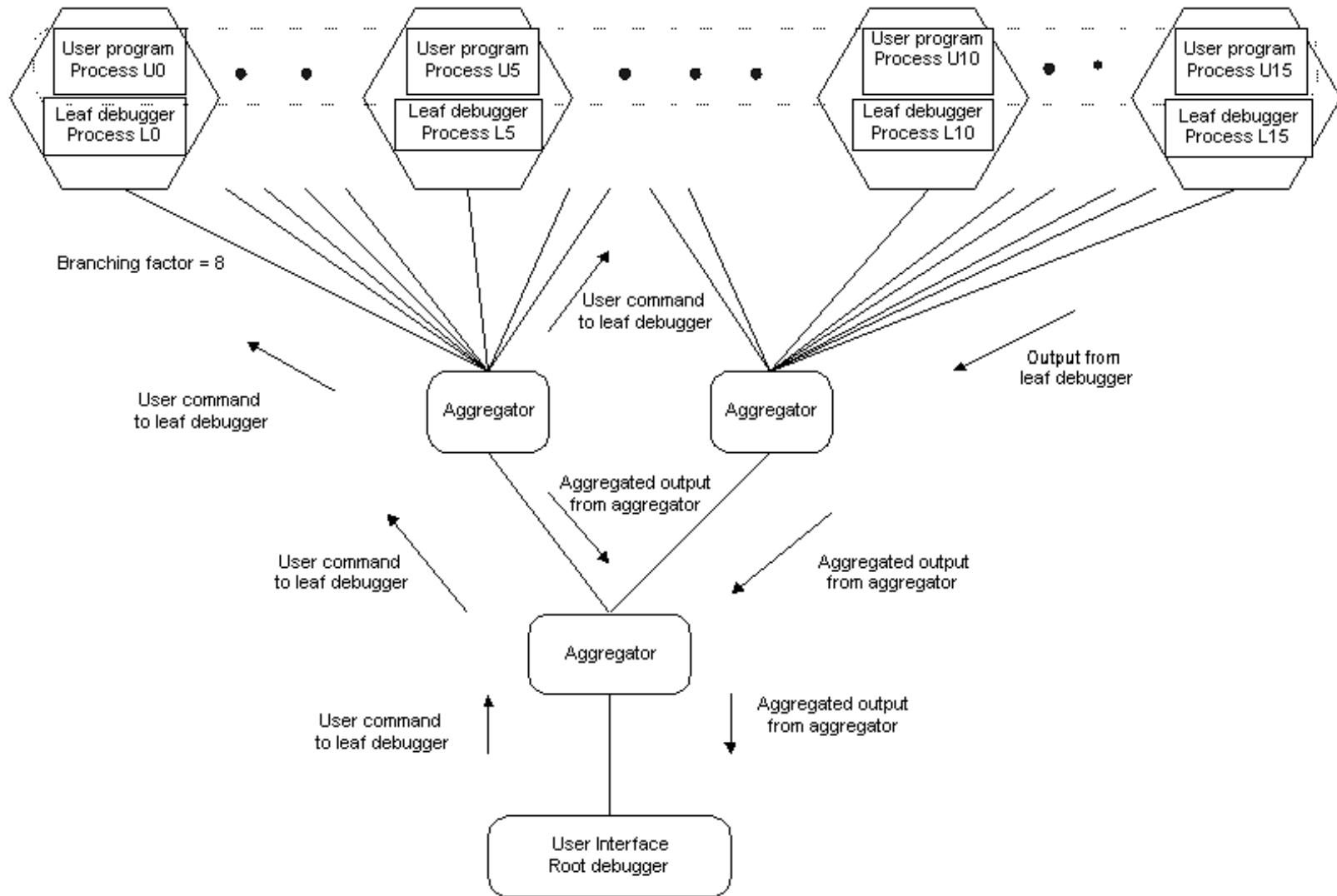
# Introduction

- Motivation
- Debugging applications on MPPs
- Traditional approach to parallel debugging
- Our approach

# Traditional debugger architecture



# Scalable parallel debugger architecture



# Aggregation of outputs

3 major categories of outputs:

1. Type 1: identical outputs from each of the debuggers

```
[0-41 ] Welcome to the Ladebug Debugger Version 65 (built Mar 20 2001 for Compaq Tru64 UNIX)
```



Process range

2. Type 2: identical outputs apart from containing different numbers

```
%1 [0-41] >2 0x120006d6c in feedback(myid=[0;41], np=42, name=0x11ffe018="mytest") mytest.c":41
```



Process range



value range

3. Type 3: widely differing outputs

# Example of outputs

```
[0] >2 0x120006d6c in feedback(myid=0 np=42, name=0x11ffe018="mytest") mytest.c":41
[1] >2 0x120006d6c in feedback(myid=1, np=42, name=0x11ffe018="mytest") mytest.c":41
[2] >2 0x120006d6c in feedback(myid=2, np=42, name=0x11ffe018="mytest") mytest.c" :41
[3] >2 0x120006d6c in feedback(myid=3, np=42, name=0x11ffe018="mytest") mytest.c":41
[4] >2 0x120006d6c in feedback(myid=4, np=42, name=0x11ffe018="mytest") mytest.c":41
.
.
[41] >2 0x120006d6c in feedback(myid=4, np=42, name=0x11ffe018="mytest") mytest.c":41

[>2 0x120006d6c in feedback(myid=?, np=42, name=0x11ffe018="mytest") "mytest.c":41
```

# Extending a traditional debugger

- Notion of sets of processes
- New Ladebug commands
  - *focus* ( *focus 5* )
    - Selects the processes that commands are sent to.
  - *show aggregated message*
  - *expand aggregated message*

# Example of user session: focus command

```
16: (ladebug) focus [0:2]
17: [0:2]>
18: [0:2]> cont
19: [0:2]> [0:2] [1] stopped at [int feedbackToDebugger(int, int, char*):18
    0x120001818]
20: [0:2] 18 int i = 0;
21: [0:2]> where
22: [0:2]> %1 [0:2] >0 0x120001818 in feedbackToDebugger(myid=[0:2], np=6,
    name=0x11fffe060="/usr/users/parallel/examples/cpi-DmpirunStop") "cpi-user1.c":18
23: [0:2] #1 0x120001a48 in main(argc=1, argv=0x11fffc018) "cpi-user1.c":63
24: [0:2] #2 0x120001758 in __start(...) in /usr/users/parallel/examples/cpi-DmpirunStop
```

# Example: show aggregated message

43: [0:5]> **show aggregated message**

44: %1 [0:2] >0 0x120001818 in feedbackToDebugger(myid=[0;2], np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-DmpirunStop") "cpi-  
user1.c":18

45: %2 [3:5] >0 0x120001818 in feedbackToDebugger(myid=[3;5], np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-DmpirunStop") "cpi-  
user1.c":18

46: %3 [0:5] >0 0x12000181c in feedbackToDebugger(myid=[0;5], np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-DmpirunStop") "cpi-  
user1.c":20

# Example: expand aggregated message

47: [0:5]> **expand aggregated message 3**

48: %3 [0:5] >0 0x12000181c in feedbackToDebugger(myid=[0;5], np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-DmpirunStop") "cpi-user1.c":20

49: [0] >0 0x12000181c in feedbackToDebugger(myid=0, np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-DmpirunStop") "cpi-user1.c":20

50: [1] >0 0x12000181c in feedbackToDebugger(myid=1, np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-DmpirunStop") "cpi-user1.c":20

51: [2] >0 0x12000181c in feedbackToDebugger(myid=2, np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-DmpirunStop") "cpi-user1.c":20

52: [3] >0 0x12000181c in feedbackToDebugger(myid=3, np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-S-BuildDmpirunStop") "cpi-  
user1.c":20

53: [4] >0 0x12000181c in feedbackToDebugger(myid=4, np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-S-BuildDmpirunStop") "cpi-  
user1.c":20

54: [5] >0 0x12000181c in feedbackToDebugger(myid=5, np=6,  
name=0x11fffe060="/usr/users/parallel/examples/cpi-S-BuildDmpirunStop") "cpi-  
user1.c":20

# Experimental results

```
>$LADEBUB_HOME/ladebug -parallel /usr/bin/prun -n 2048 ~/smg98_work/test/smg98
```

```
Welcome to the Ladebug Debugger Version 67 (built Dec 18 2001 for Compaq Tru64 UNIX)
```

```
Reading symbolic information ...done
```

```
stopped at [void _rms_breakpoint(void):1616 0x3ffbff69750]
```

```
[snip]
```

```
(ladebug) [0:2047] Welcome to the Ladebug Debugger Version 67 (built Dec 18 2001 for Compaq Tru64 UNIX)
```

```
[0:2047] -----
```

```
[0:2047] object file name: /usr/users/7/balle/smg98_work/test/smg98
```

```
[0:2047] Reading symbolic information ... [0:2047] done
```

```
[0:2047] 53 int A_num_ghost[6] = { 0, 0, 0,0, 0, 0};
```

```
(ladebug) next
```

```
(ladebug) [0:2047] stopped at [<opaque> __start(...) 0x120006468]
```

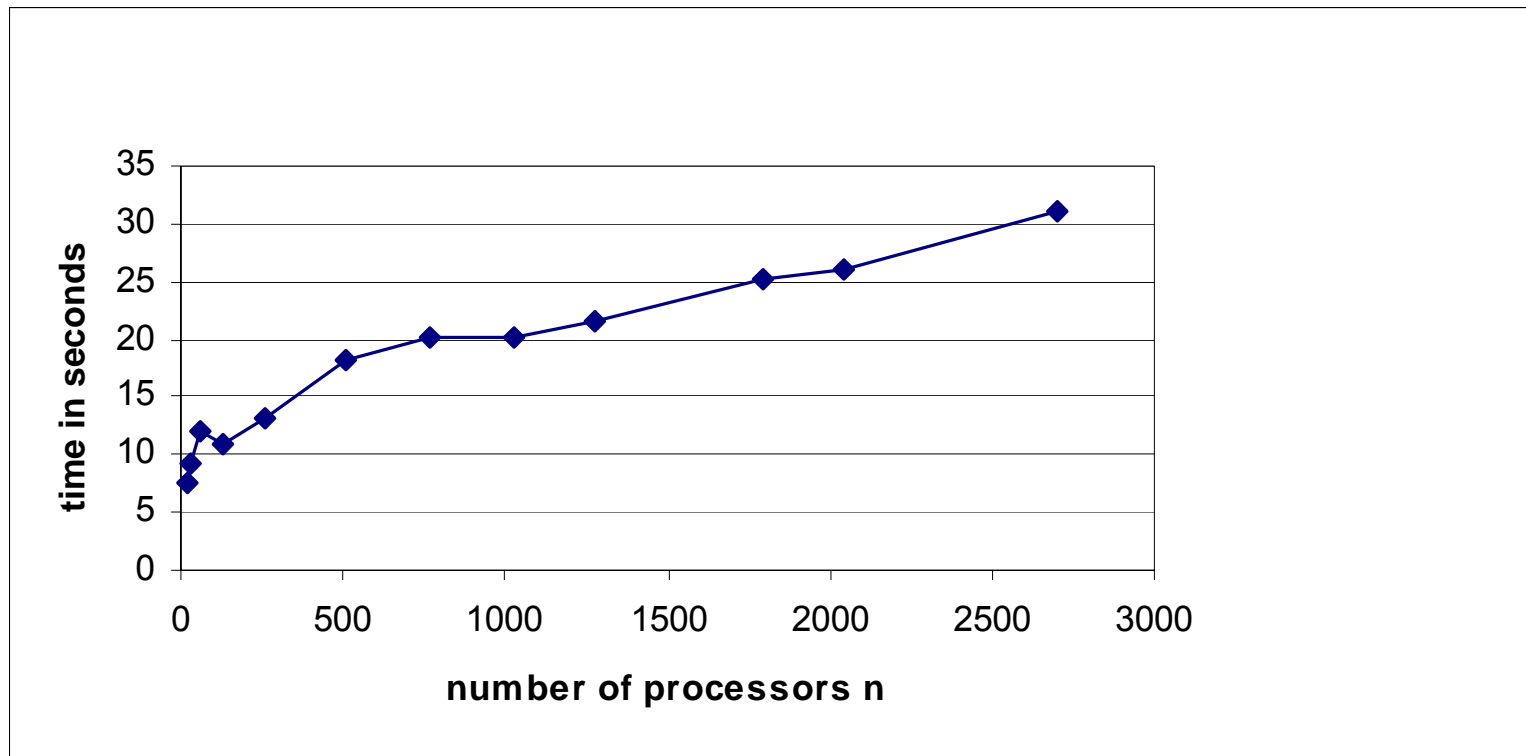
```
[0:2047]
```

```
[0:2047] Information: An <opaque> type was presented during execution of the previous command.  
For complete type information on this symbol, recompilation of the program will be necessary.  
Consult the compiler man pages for details on producing full symbol table information using the '-g' (and '-gall' for cxx) flags.
```

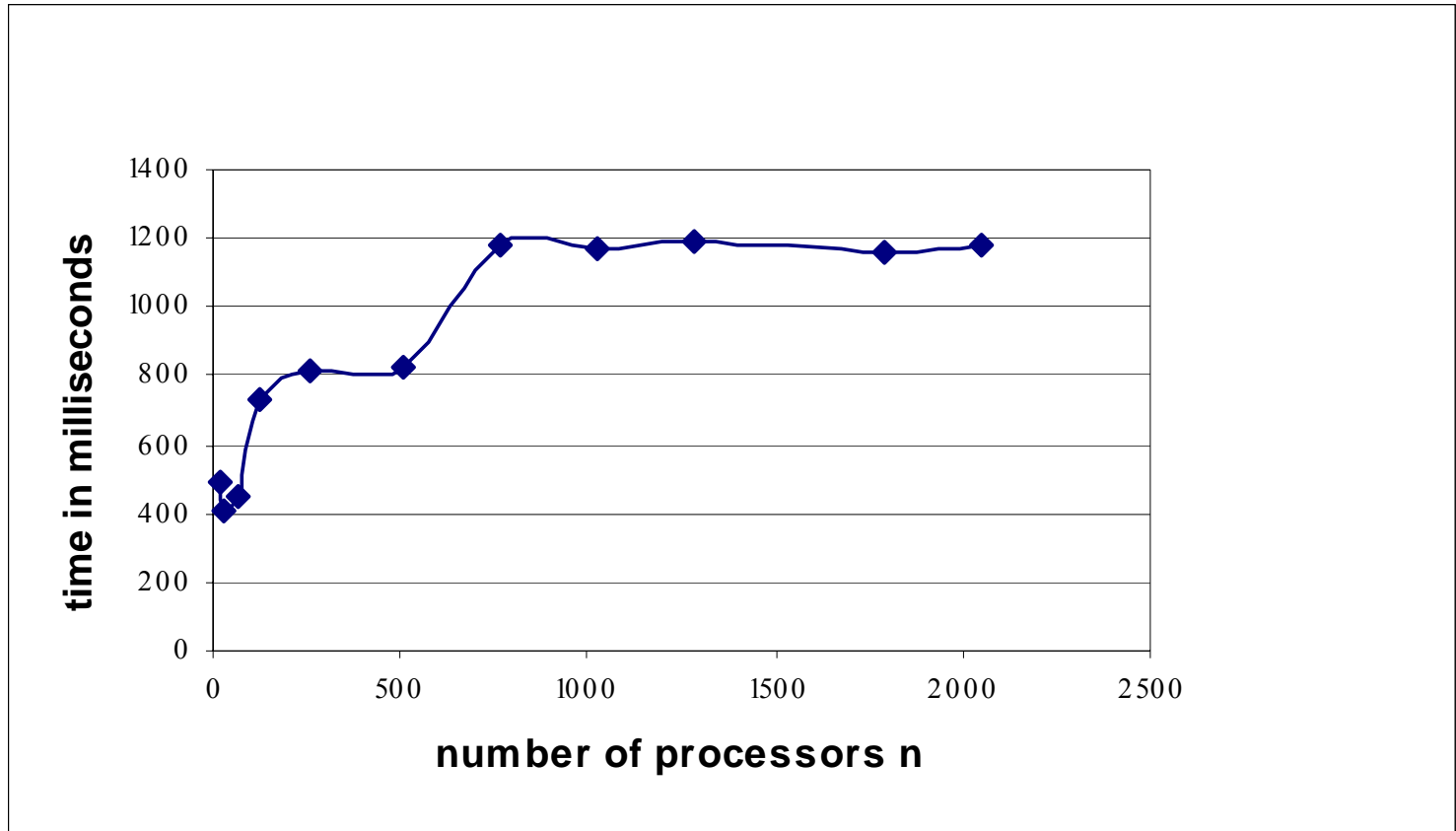
```
[0:2047]
```

```
(ladebug)
```

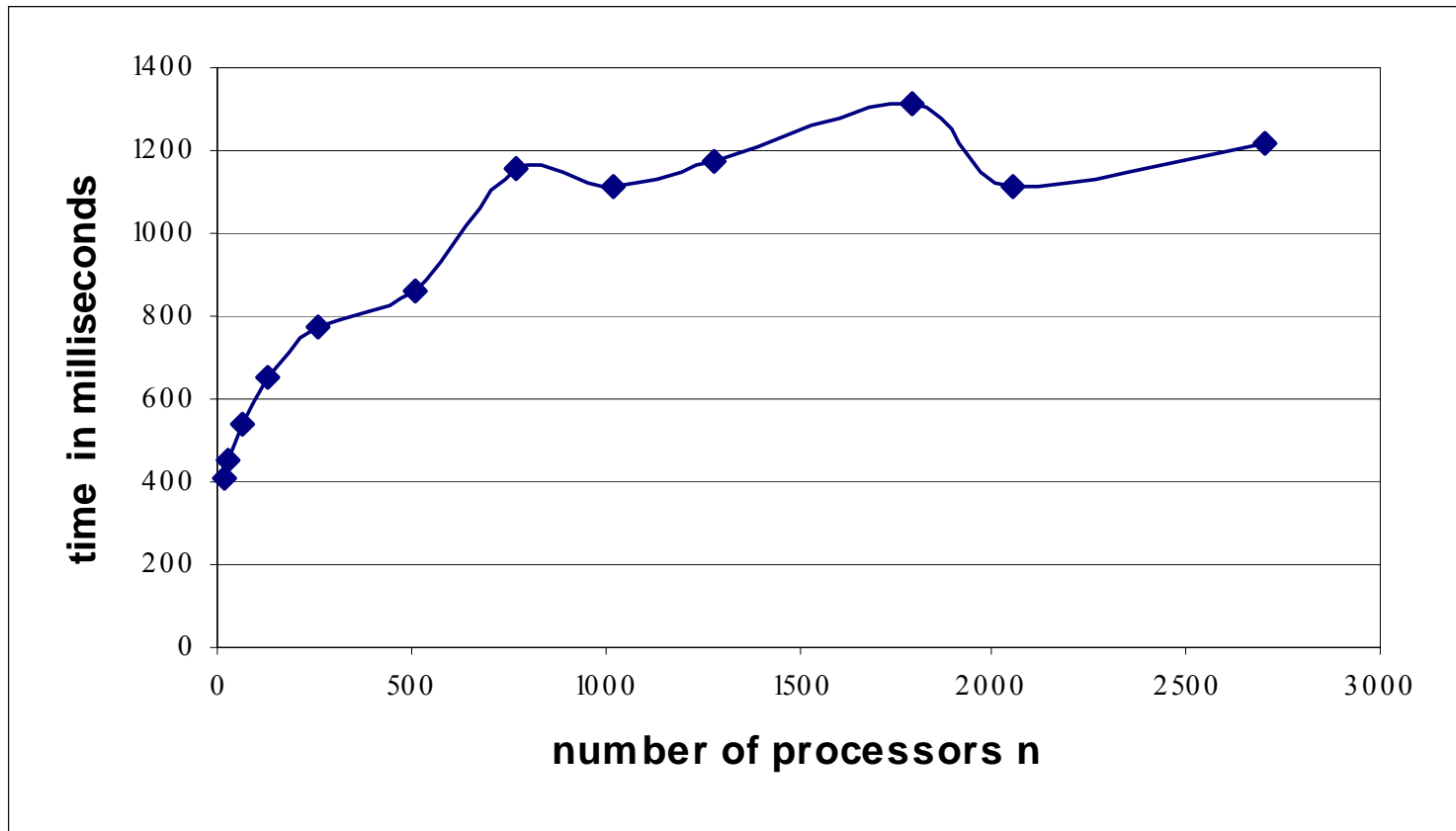
# Debugger startup



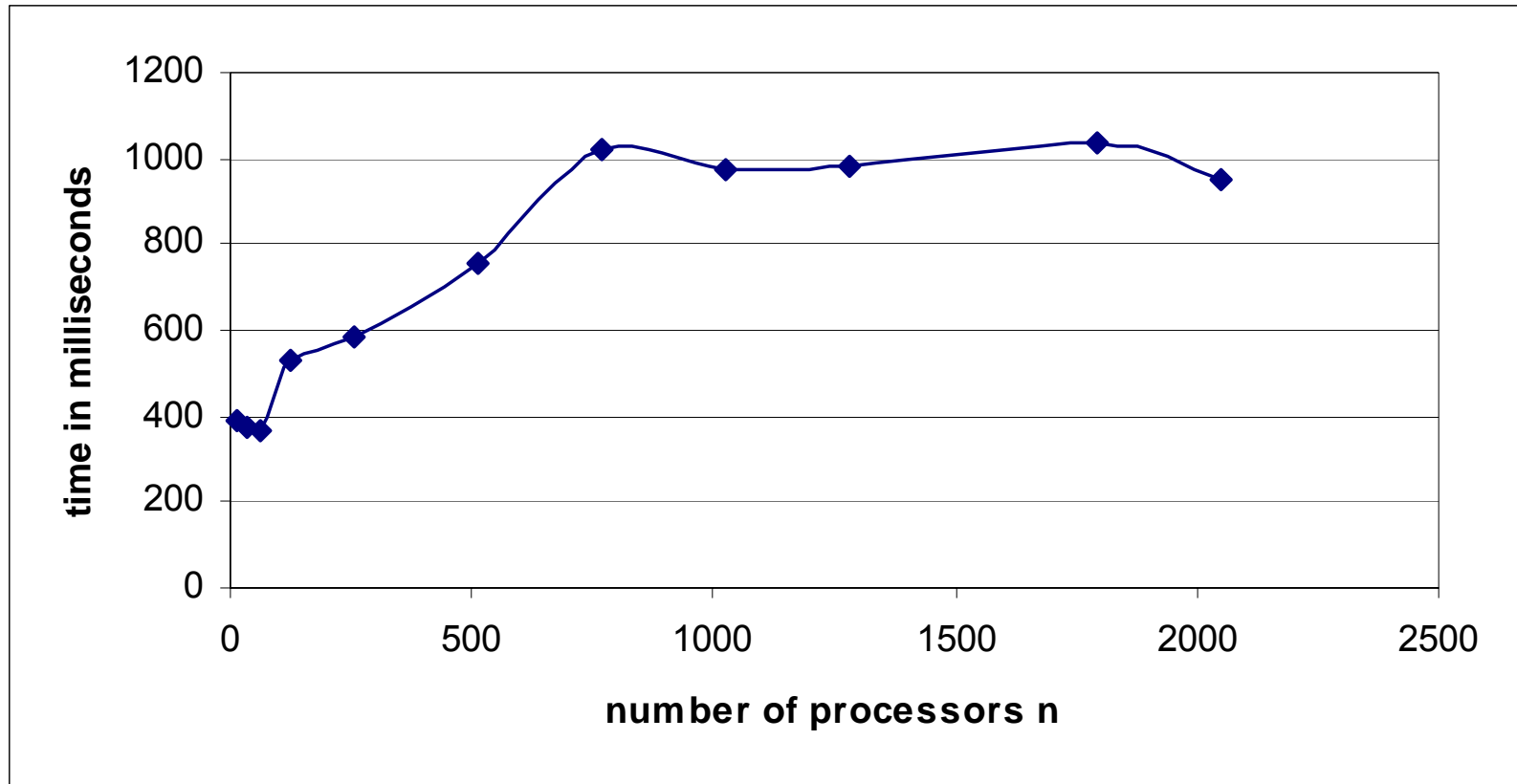
# Stop command



# Next command



# Where command



# Final remarks

- Scalable architecture
- User friendly debugger
- Good results
  - Proof of concept
- Extended the HPDF functionality
  - Aggregation of type 2 output
  - Added new commands
    - Show aggregated message
    - Expand aggregated message
- Ready to be release with Ladebug 4.0-67 in March 2002

# Acknowledgements

- Alex Holmansky, Gordon Saladino, John Bishop, Julie Goonan
- Steve Tolnai, Siew Ching Tan (VPAC), Steve Quenette (VPAC)
- Phil Becker
- J. Ray Scott and Sergiu Sanielevici (Pittsburgh Supercomputing Center)
- Michelle Trull