

Demonstration of Interactive Runtime Debugging of Distributed Dataflows in Texera

Zuozhi Wang, Avinash Kumar, Shengquan Ni and Chen Li

UCI

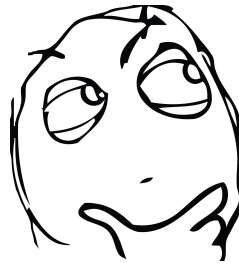
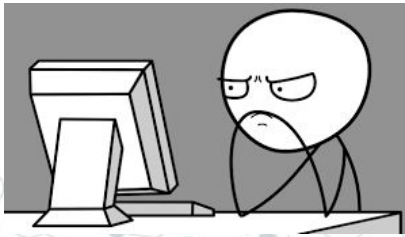
University of California, Irvine

Big data analytics workflows

Problems in current systems:

Little Feedback

No Runtime Debugging



printf() vs gdb

```
satapouch@satapouchPC ~/Stažené $ cat boot.log
Begin: Loading essential drivers ... done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
Begin: Running /scripts/local-premount ... done.
Begin: Running /scripts/local-bottom ... done.
done.
Begin: Running /scripts/init-bottom ... done.
* Stopping load modules from /etc/modules [ OK ]
* Stopping cold plug devices [ OK ]
* Stopping log initial device creation [ OK ]
* Starting enable remaining boot-time encrypted block devices [ OK ]
* Stopping Read required files in advance [ OK ]
* Starting Mount filesystems on boot [ OK ]
* Stopping Track if upstart is running in a container [ OK ]
* Starting Signal sysvinit that virtual filesystems are mounted [ OK ]
* Starting Signal sysvinit that virtual filesystems are mounted [ OK ]
* Starting Signal sysvinit that remote filesystems are mounted [ OK ]
* Starting Signal sysvinit that the rootfs is mounted [ OK ]
* Starting Clean /tmp directory [ OK ]
* Stopping Clean /tmp directory [ OK ]
* Starting SMB/CIFS File Server [ OK ]
* Starting Signal sysvinit that local filesystems are mounted [ OK ]
* Starting restore software rkill state [ OK ]
* Starting configure network device security [ OK ]
* Stopping restore software rkill state [ OK ]
* Stopping Mount filesystems on boot [ OK ]
* Starting flush early job output to logs [ OK ]
```

Hard to track information

The screenshot shows the Eclipse IDE interface for a C++ project named 'NonStop'. The main editor displays the source code of 'NonStop.cpp', which includes a loop that prints 'another loop\n' and creates three threads. A red box highlights the printf statement at line 44, with a note 'Line where the trace was collected'. The Variables view on the right shows the state of variables for the selected thread (thread3), including 'message3' (0x8048821) and 'iret3' (0). A red box highlights the 'message3' value with the text 'Collected Data'. The Breakpoints view shows a breakpoint at line 41, with a note 'Tracepoint for the selected trace'. The Trace Control view shows the current trace frame (frame 4) and options for 'Stop visualization' and 'Change trace'.

Real-time Interaction
Easy to understand

Full Paper:

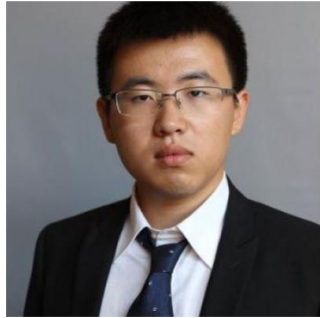
Amber: A Debuggable Dataflow System Based on the Actor Model

- Avinash Kumar, Zuozhi Wang, Shengquan Ni, Chen Li.
VLDB 2020

[03A] Analytical Query Processing 1

[52A] Analytical Query Processing 1 (repeat)

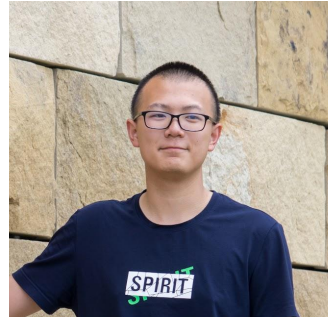
Demonstration of Interactive Runtime Debugging of Distributed Dataflows in Texera



Zuozhi Wang



Avinash Kumar



Shengquan Ni



Chen Li

UCI

University of California, Irvine