



Toward a Data-Centric Profiler for PGAS Applications

Hui Zhang, Jeffrey K. Hollingsworth(advisor)

{hzhang86, hollings}@cs.umd.edu

Department of Computer Science, University of Maryland-College Park

Why we need a data-centric profiler ?

```
1: int busy(int *x) {  
2:   *x = complex(); //consumes the most time  
3:   return *x;  
4: }  
5: int main() {  
6:   for (i=0; i<n; i++) {  
7:     A[i] = busy(&B[i]) + busy(&C[i]);  
8:   }  
9: }
```

Code-centric Profiling

main: 100% latency
busy: 100% latency
complex: 100% latency

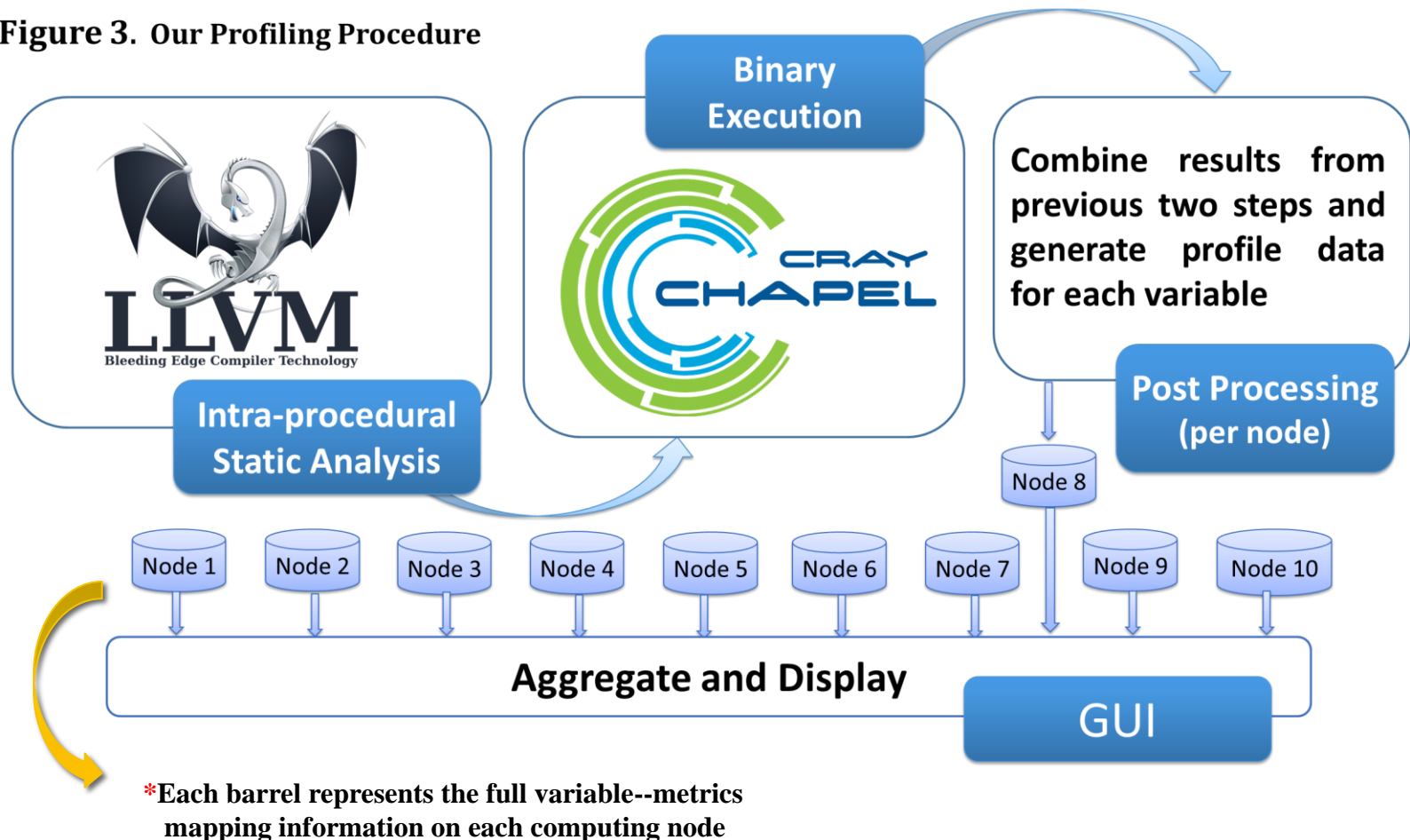
Data-centric Profiling

Array A: 100% latency
main
Array B: 50% latency
main.busy.complex
Array C: 50% latency
main.busy.complex

Figure 1. Code-centric aggregates metrics to the different functions based on sampled lines, while data-centric can distinguish these metrics by different variables

How the tool works ?

Figure 3. Our Profiling Procedure



A Simple Test Program

```
record Birthday {  
    var year: int;  
    var month: int;  
    var day: int;  
}  
  
record Actor {  
    var name:string;  
    var bd:Birthday;  
}  
  
for i2 in 1..LARGE {  
    mid = i%8;  
    ActorA.bd.month = CPUheavier(mid);  
    ActorA.bd.day = CPUheavy(mid);  
}  
for i in 1..LARGE/2 {  
    ActorB.bd.year = CPUheavy(i);  
}
```



Full Code Centric	
Elame Points (Data/Code Hybrid)	
Full Data Centric	
Program Blame	
ActorA [Actor]	main.main 79.17%
bd [Birthday]	main.main 79.17%
month [Int]	main.main 55.95%
day [Int]	main.main 40.48%
Number [Int]	main.main 40.48%
ActorB [Actor]	main.main 20.83%
bd [Birthday]	main.main 20.83%
year [Int]	main.main 20.83%
mid [Int]	main.main 17.26%
localVar [Double]	main.main 3.57%
i2 [Int]	main.main 2.38%
local_LARGE [Int]	main.main 1.79%
i [Int]	main.main 1.79%

Figure 4. Example Chapel code and profiling result

THANK YOU !

