



CORSE: COMPILER OPTIMIZATION AND RUN-TIME SYSTEMS *

Fabrice Rastello

* INRIA Joint Project Team

Workshop EnergySFE. Grenoble. September 2016





Outline

Compiler Optimization and Runtime SystEms



- 1 Project-team composition / Institutional context / Objectives
- 2 Compilers should be hybrid
- 3 Runtime should be interfaced with compilers
- 4 Debuggers should be aware of HL Prog. info



Outline

Compiler Optimization and Runtime Systems



- 1 Project-team composition / Institutional context / Objectives
- 2 Compilers should be hybrid
- 3 Runtime should be interfaced with compilers
- 4 Debuggers should be aware of HL Prog. info



Institutional context / Objectives

Compiler Optimization and Runtime SystEms



Joint Project-Team in the LIG laboratory @ GIAN/Minatec

- Fabrice RASTELLO, Florent BOUCHEZ TICHADOU, François BROQUEDIS, Frédéric DESPREZ, Yliès FALCONE, Jean-François MÉHAUT
- 8 PhD, 2 Post-doc, 2 Engineer



inria
informatics mathematics



Domain : Compiler optimization and runtime systems for **performance** and energy consumption (not reliability, nor WCET)

Issues: Scalability and heterogeneity/complexity $\equiv$ trade-off between specific optimizations and programmability/portability

Applications: loop-nests / graph-algorithmic / signal-processing / geophysics / nano-materials

Approach: **combine** static/dynamic approaches & compiler/run-time-level solutions

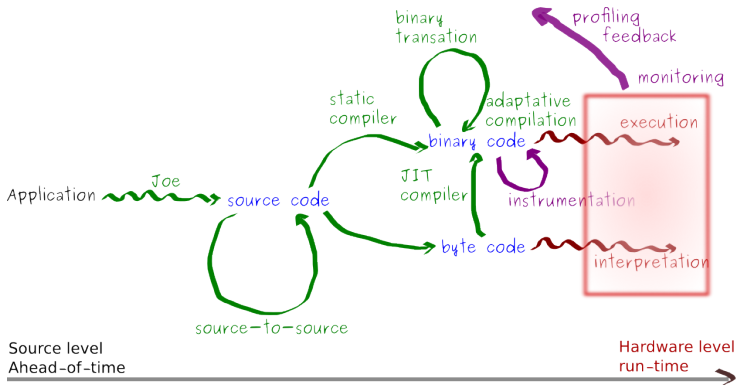


- 1 Project-team composition / Institutional context / Objectives
- 2 **Compilers should be hybrid**
- 3 Runtime should be interfaced with compilers
- 4 Debuggers should be aware of HL Prog. info



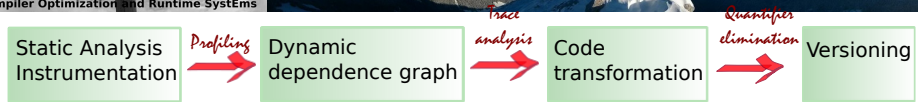
Compiler technologies today

Compiler Optimization and Runtime Systems



Static/Dynamic Analysis: Example

Compiler Optimization and Runtime Systems



dependencies

```
for ( i=1 ; i<n ; i++ )  
  for ( j=1 ; j<m ; j++ )  
    t[8*H*j+8*i] =  
      t[-8*H+8*H*j+8*i]  
      + t[-8+8*H*j+8*i];
```

run-time dependence profiling

- (i,j) depends on $(i-1,j)$
and on $(i,j-1)$
- **only** if $n \leq H$
- **conditional** loop interchange
(cloning) $\times 5$ faster



Outline

Compiler Optimization and Runtime SysEms



- 1 Project-team composition / Institutional context / Objectives
- 2 Compilers should be hybrid
- 3 Runtime should be interfaced with compilers
- 4 Debuggers should be aware of HL Prog. info

The role of the Runtime

Compiler



?

Compiler output : A binary (.exe) file

Compiler adds calls to the runtime routines

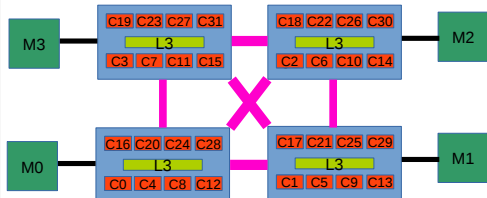
Runtime

Setup Parallel Execution (Mem)

- Static Data
- Dynamic Heap

Parallel Execution (Core)

- Task/Actor/Chare
- Synchro/Comm
- Load Balancing



Interfaced run-time with compilers: Illustrating

Compiler Optimization and Runtime Systems

$b \times b \times b$ Blocked matrix-matrix multiply

```
for (I=0; I<n; I+=b)
  for (J=0; J<n; J+=b)
    for (K=0; K<n; K+=b)
      C[I..I+b, J..J+b] += A[I..I+b, K..K+b] * B[K..K+b, J..J+b]
```

Static Complexity Analysis

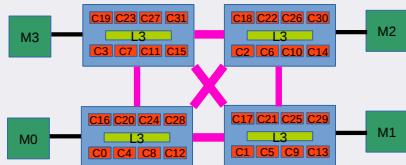
- Arithmetic: $F = 2n^3$
- Memory-access (cache size S)

$$\begin{aligned} n^2 \ll S & : Q = 3n^2 \\ n \times b \ll S & : Q = n^3/b \\ b \times b \ll S & : Q = 2n^3/b \\ b \ll S & : Q = n^3 \end{aligned}$$

Run-time

Load balancing:

- CPU, Mem bandwidth/footprint





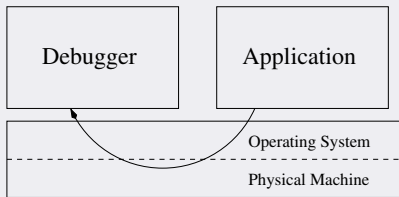
- 1 Project-team composition / Institutional context / Objectives
- 2 Compilers should be hybrid
- 3 Runtime should be interfaced with compilers
- 4 Debuggers should be aware of HL Prog. info



Debugging

Finding bugs/defects with software/system

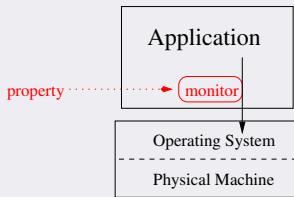
- ⊕ low-level
- ⊖ low-level
- ⊖ tedious, no guidance



Monitoring

Checking program execution against properties

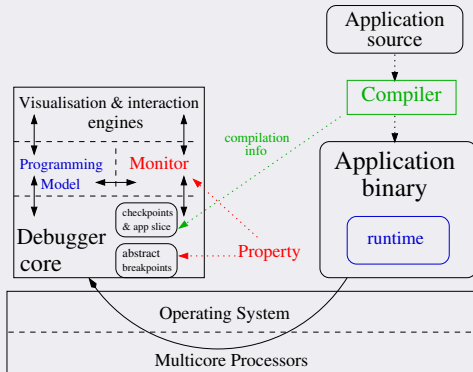
- ⊕ formal-method, abstraction
- ⊖ limited adoption
- ⊖ instrumentation (capture events)



Unified Functional Debugging/Verification

Compiler Optimization and Runtime Systems

Analyse; Debug/Monitor



OpenMP code with deadlock

```
void f(int r) { /*...*/  
    if ( r == 0 )  
        #pragma omp barrier  
    return; }  
  
int main ( ) {  
    int r;  
    #pragma omp parallel private(r)  
    {  
        r = ...;  
        f( r );  
        #pragma omp barrier  
        /*...*/  
    }  
    return 0; }
```



Questions?

Compiler Optimization and Runtime Systems



inria
informatics mathematics