Colorado State University

Polyhedral Model: Past Present & Future

title shamelessly copied from Feautrier's LCPC 2009 keynote

Sanjay Rajopadhye

The Past

- Automatic Parallelization
- Abstract interpretation
- Systolic Array Synthesis

Karp Miller & Winograd 1967

- Personal advice: read this paper many times, over many years

  - Proposed mathematical equations as program representation
  - Analysis to detect parallelism (schedules)
  - … and much more

Karp, Miller & Winograd, JACM 1967: "The Organization of Computations for Uniform Recurrence Equations"

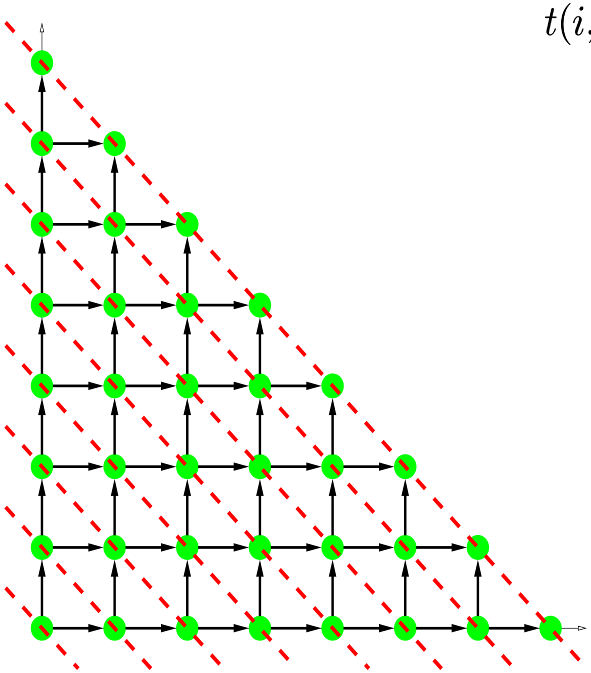
KMW67: Scheduling a URE

- Domain of the equation: entire positive quadrant/orthant (but can be more specific)
- Dependences: constant vectors
- Hyperplane scheduling: Find λ such that all points $\lambda z = t$ have timestamp t. λ is the normal vector of the schedule hyperplane

Do i, j | 0;<= (i, j) < N}

Example

$$X[i,j] \;=\; g(X[i-1,j], X[i,j-1]$$
$$t(i,j) \;\equiv\; ai + bj + \alpha$$



Schedule validity conditions

$$[a,b][0,-1]^T \;<\; 0$$
$$[a,b][-1,0]^T \;<\; 0$$
$$\alpha \;\geq\; 0$$

i.e., $\{a, b, \alpha \mid a, b > 0, \alpha \geq 0\}$

Optimal schedule: $t(i,j) = i + j$

What more do you need?

- Systolic Array Synthesis (single UREs)
  - Find a schedule
  - and a processor allocation function
  - Do a "space-time transformation"
- Lamport 74 (loop parallelization)
  - in a perfectly nested loop, array variables are accessed (read/written) using a special subclass of affine functions
  - But main result is for uniform dependences

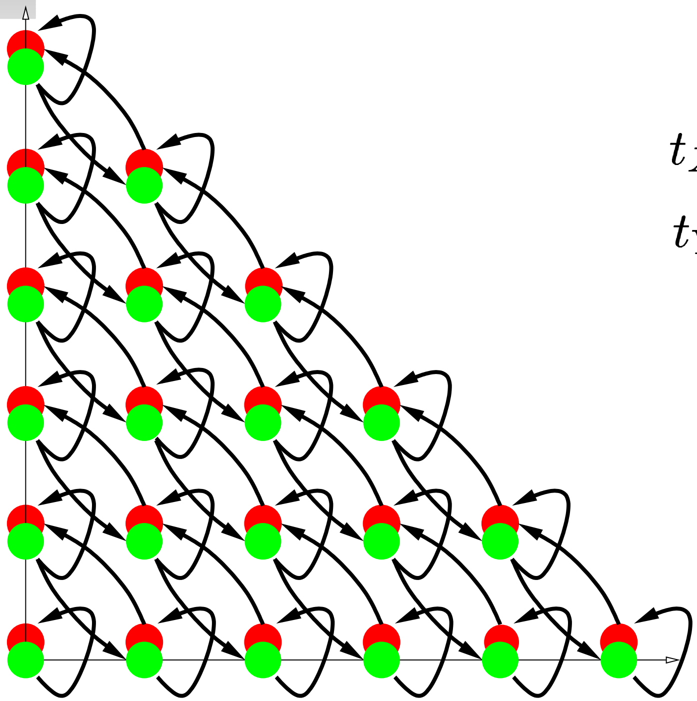
Lamport, CACM 1974: "Parallel Execution of DO Loops"

The world is not uniform

- Synthesizing Systolic Arrays from AREs
- More general loops than Lamport[1]
- Even Uniform Needs Affine

1. "It is possible generalize [to …] any linear [access] function but […] results become weaker and more complicated"

MW67: SURE *A less contrived example*

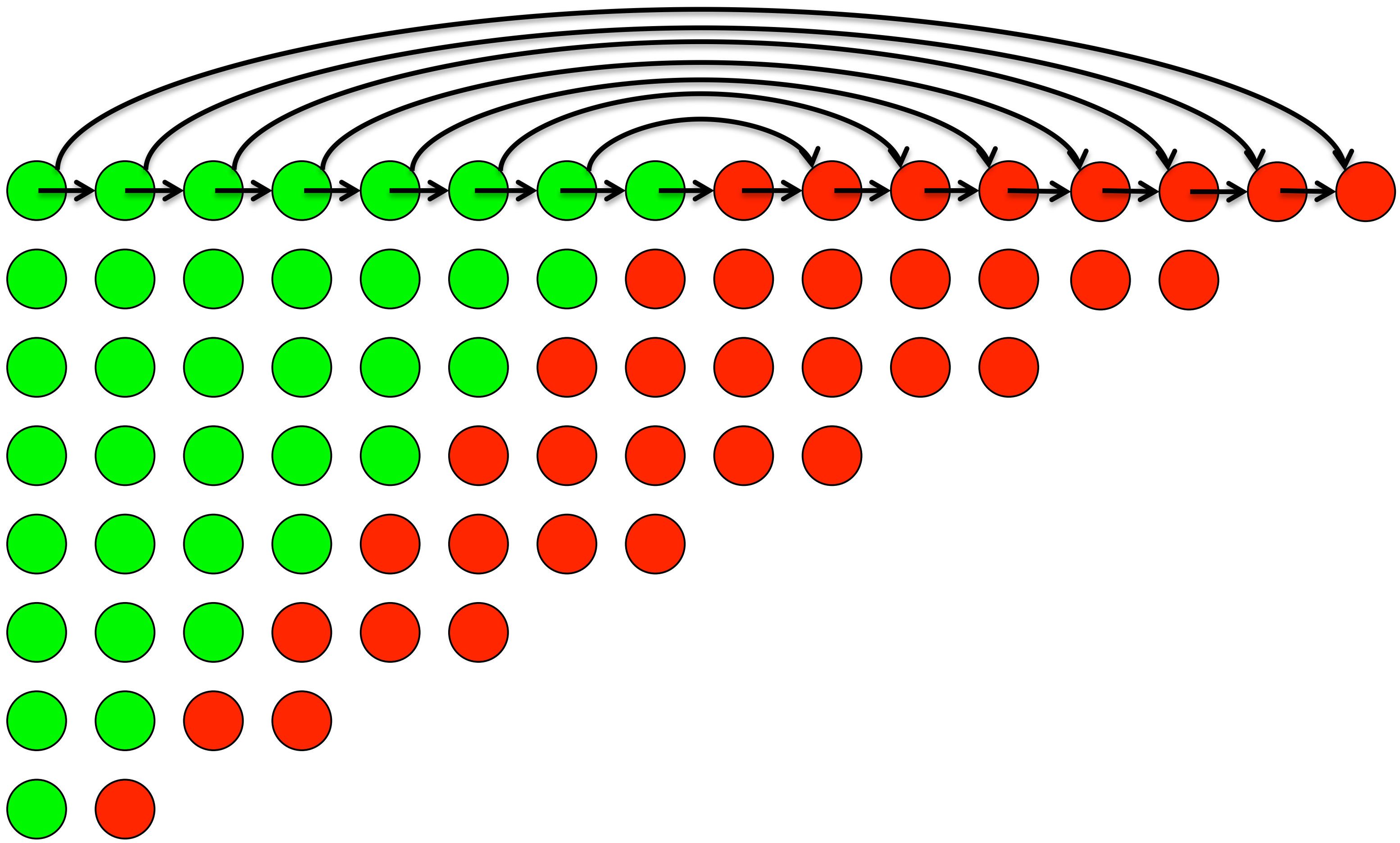


$$
\begin{aligned}
X[i,j] &= g(X[i-1,j+1] \\
Y[i,j] &= h(Y[i+1,j-1], X[i,j] \\
t_X(i,j) &= a_X i + b_X j + \alpha_X \\
t_Y(i,j) &= a_Y i + b_Y j + \alpha_Y
\end{aligned}
$$

Optimal solution

$$
\begin{aligned}
t_X(i,j) &= i \\
t_Y(i,j) &= i + 2j + 1
\end{aligned}
$$

Transformed graph



Limitations of KMW 1967

- Provides sophisticated analysis, but stops there
- Transforming an SURE with **variable dependent** schedules is not an SURE
  - Need a richer program representation
  - Need to generalize the analysis
  - Need to develop a theory of transformations
- SARE: Systems of Affine Recurrence Equations

But loops are not equations

- "Uniform" memory access functions do not necessarily imply uniform dependences
- Loop dependences are always lexicographically positive
- Is KMW overkill for loops?

Feautrier

- Affine control (mistakenly called "static" control) loops are semantically the same as SAREs
- New scheduling algorithms based on Farkas' lemma
- single and multi-dimensional time

Feaurier, IJPP1991: "Dataflow analysis of array and scalar references"
Feaurier, IJPP1992: "Some efficient solutions to the affine scheduling problem: Parts I and II"

Polyhedral Model: essentials

- Polyhedral program representation:
  - SAREs, Affine Control Loops (ACLs) etc.
  - Closure Properties
- Polyhedral Analysis (optimization & tools)
  - LP/ILP/PIP
  - Scheduling, (processor, memory) allocation
  - Nonlinear optimization
- Polyhedral Transformations (CoB, Index set Splitting, etc.)
- Code generation (getting out of the model)

Dependence Analysis

- Irigoin
- Allen-Kennedy
- Banerjee
- Feautrier
- Pugh
- Creusillet

Polyhedral Analysis

- Scheduling
  - Darte, Vivien, Robert, Quinton, Saouter
- Processor allocation (distribution)
  - Systolic community, Feautrier, Dion, Robert, Li, Chen
- Synchronization
  - Lim, Lam
- Memory & counting
  - Feautrier, Lefebvre, Padua, Maydan, Quilleré, Rajopadhye, Clauss

Code Generation

- From single polyhedra
  - Ancourt, Irigoin, LeVerge, Wilde, LeFur, Chamski
- From unions of polyhedra
  - Pugh, Rosser, Greibl, Lengauer, Wilde, Quilleré, Rajopadhye, Bastoul, Feautrier, Boulet
- Tiled code generation
  - Bastoul, Sadayappan, Hartono, Bhaskaran Renaganarayan, Rajopadhye, Kim,

Tiling: polyhedral model meets its Waterloo

- Non-linear transformation – breaks closure properties
- A (**the most**) critical transformation
- A long and rich history (analysis problem)
  - Systolic synthesis
    - Darte, Delosme, Fortes, Teich, Thiele, Bu, Deprettere
  - Compiling for parallelism
    - Irigoin, Schreiber-Dongarra, Ram-Saday, Darte
  - Compiling for Locality
    - Wolf-Lam,
  - Hierarchical tiling

Modern Polyhedral Model

Rubber meets the road (tools, tools, tools)

- PIPS
- PluTO
  - Bondhugula, Sadayappan, …
- Fundamental tools
  - Verdoolaege, Feautrier, Bastoul
- POCC
  - Pouchet, Vasillache, Cohen
- WHIRL/WrapIT
- Graphite, Polly, …
- High level synthesis

Conclusion

- History is in the eye of the beholder
- Sorry for any omissions