

# Getting the Most Out of Your Software Toolchain in the Era of Heterogeneous Manycore

**Nate Clark**

<http://www.cc.gatech.edu/~ntclark>

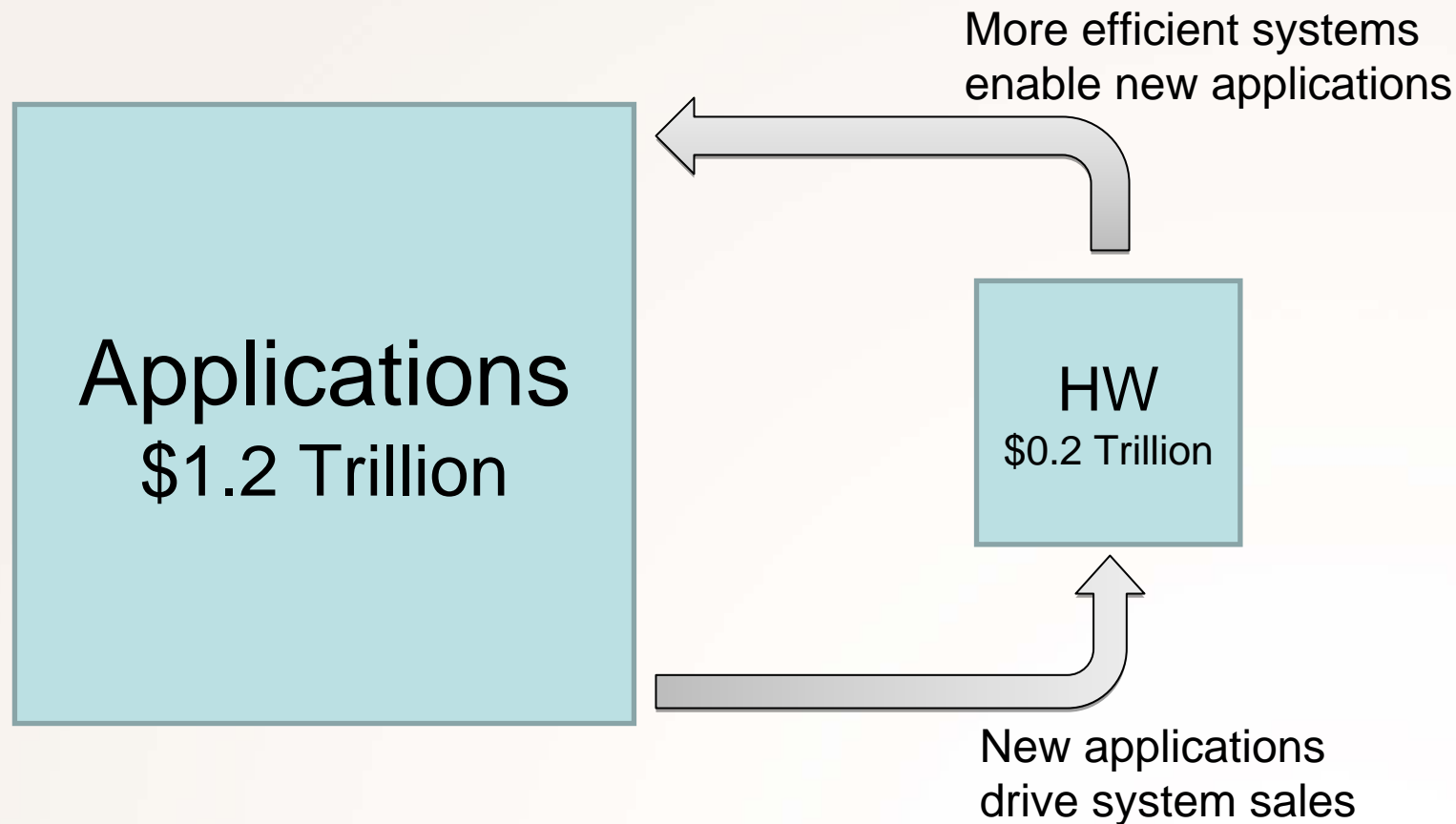


# Computational Efficiency

- New applications become possible
  - Old applications get cheaper
    - (no matter where they're executing)
- Physics → parallelism and heterogeneity
- More efficiency means software change



# Economics of Software Change



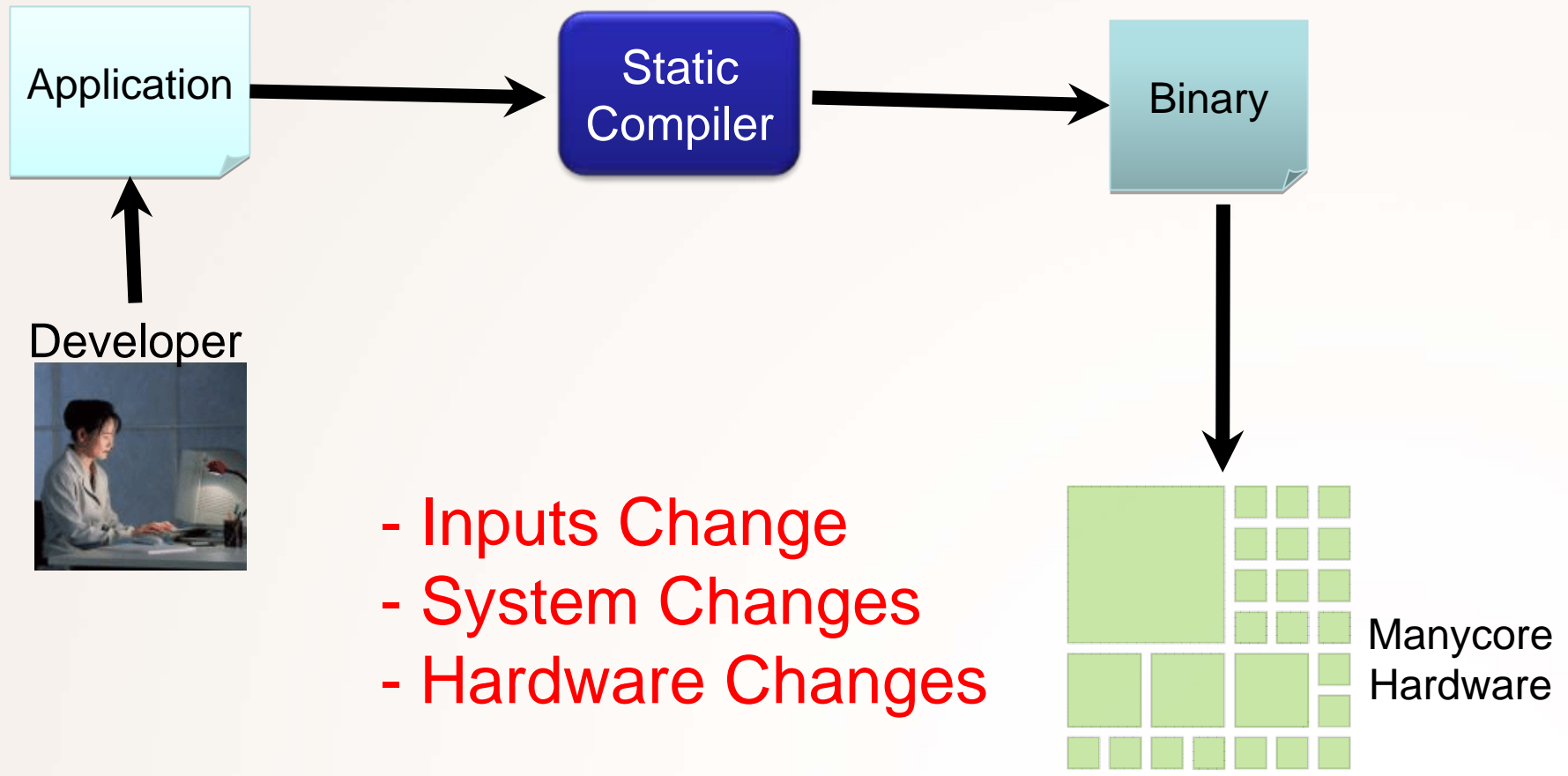


# How To Maintain Application Growth Rate

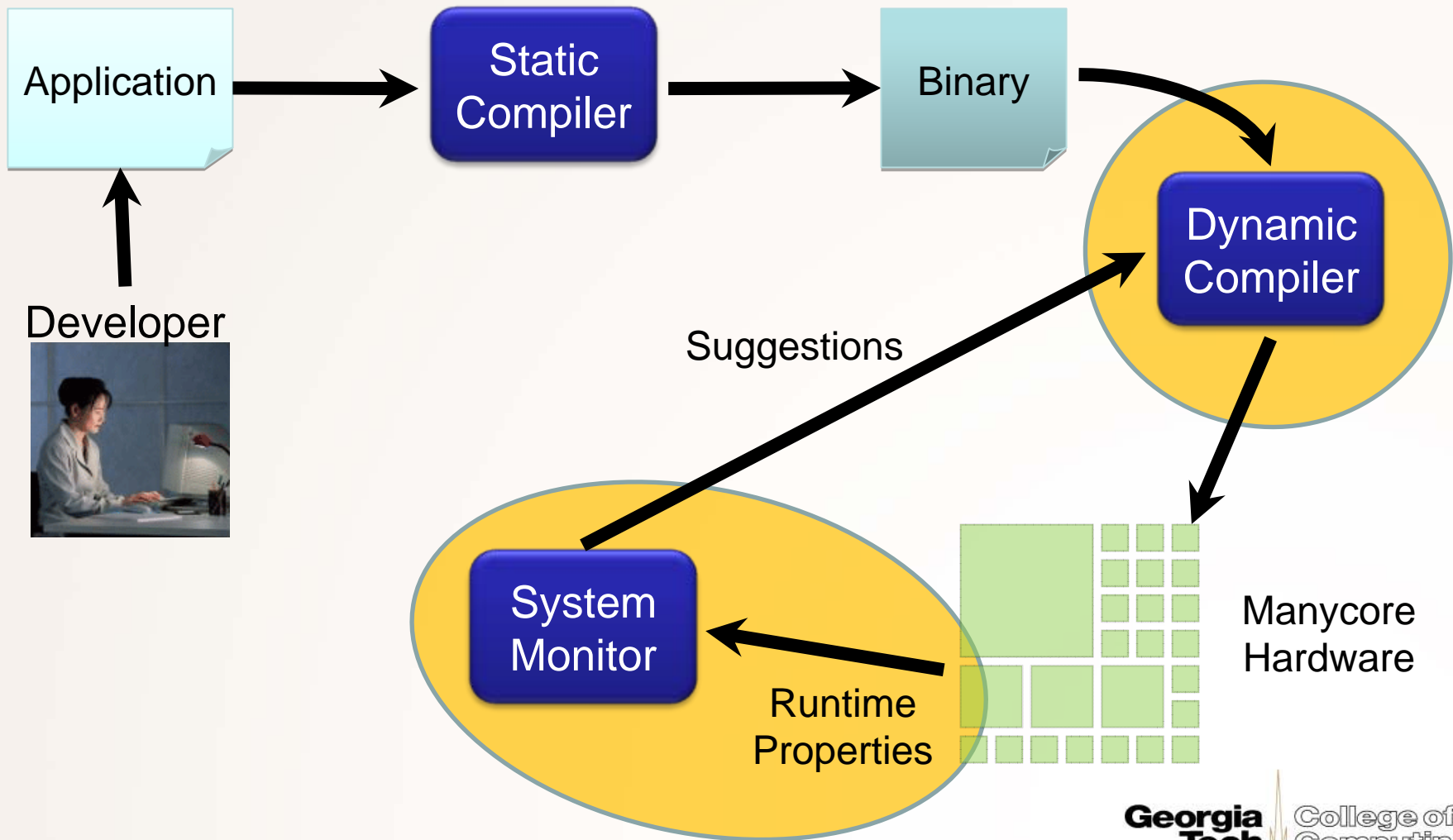
- Automate as much as possible
  - Tools to leverage parallelism/accelerators
  - Developer assistance when automation fails
- 2 Techniques
  - Dynamic program adaptation
  - Synthesizing higher-level semantics



# Why Dynamic Adaptation?

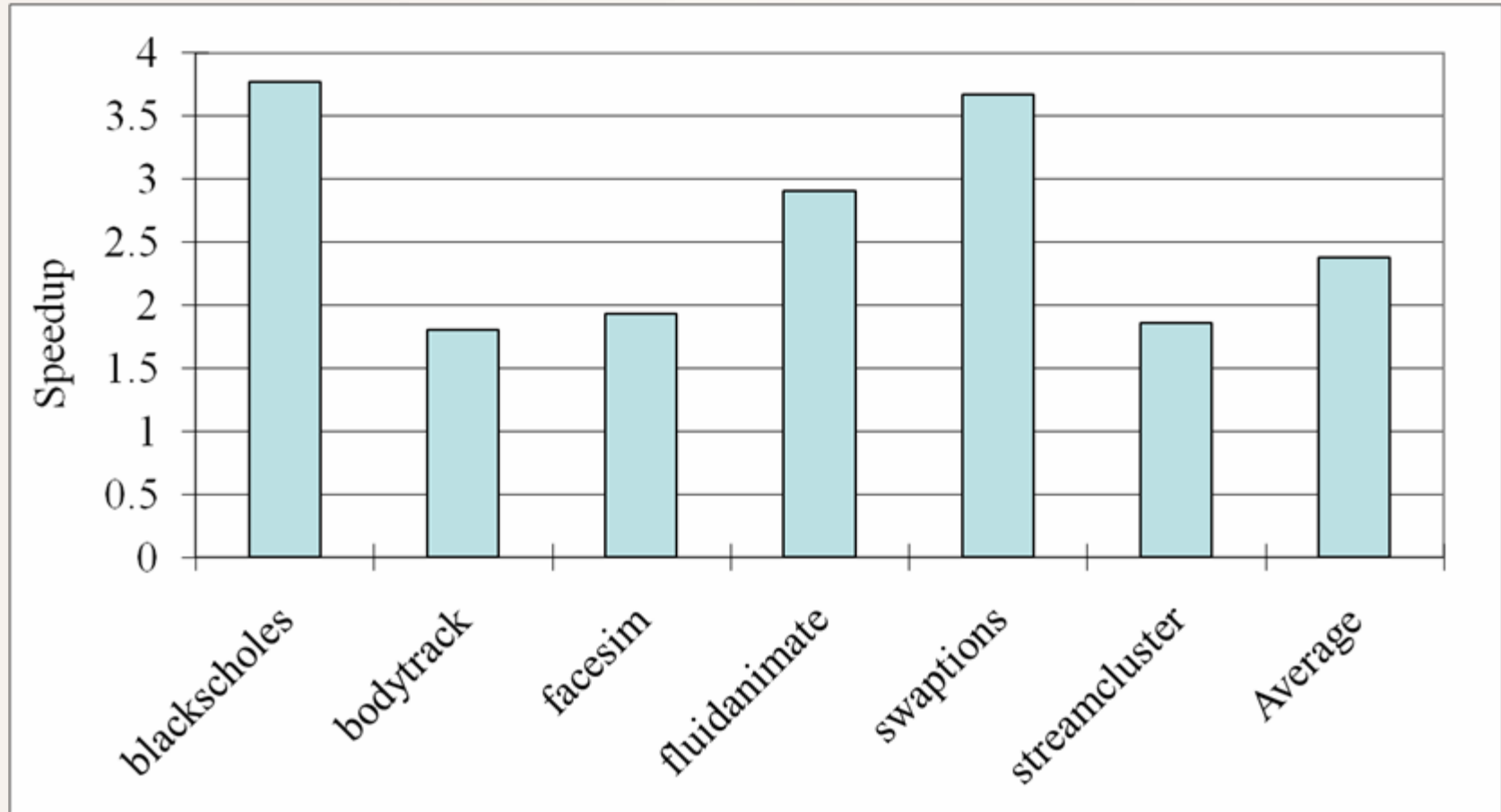


# Proposed System and Challenges





# Preliminary Results





# Synthesizing Higher Level Semantics

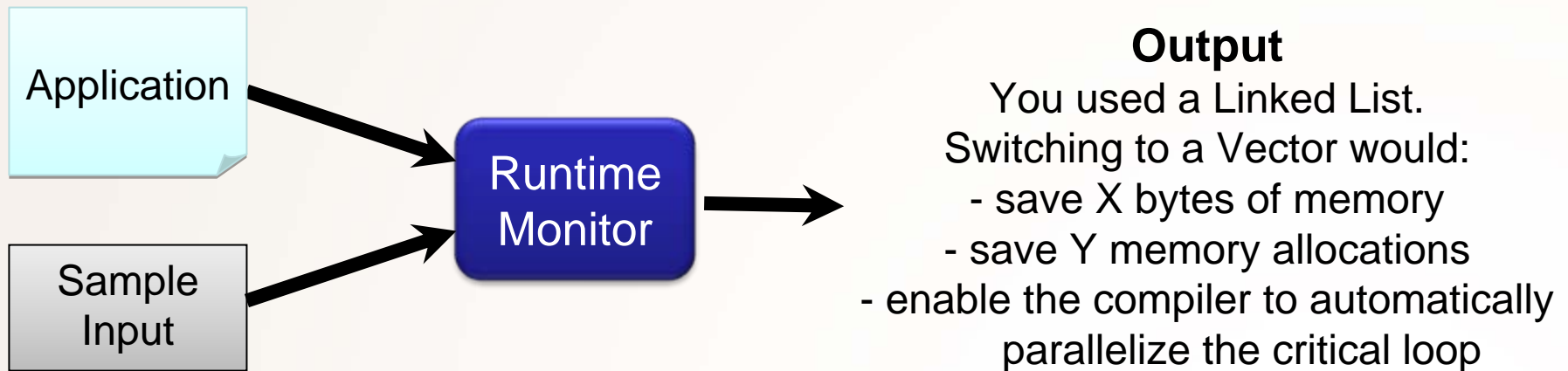
- “Decades of automatic parallelization work has been a failure...” – James Larus
- Why?
- What can humans do that machines can't?





# Example: Identifying Data Structures

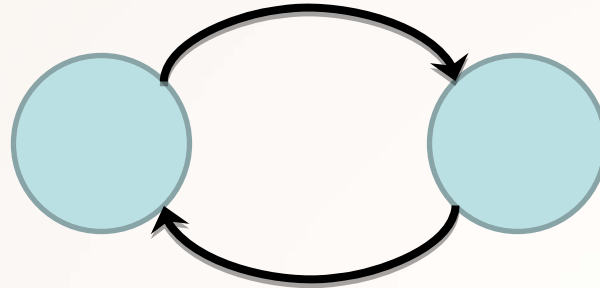
- Data structures often limit parallelism
  - Traditional program analysis looks at LD/ST
    - Misses the high-level meaning





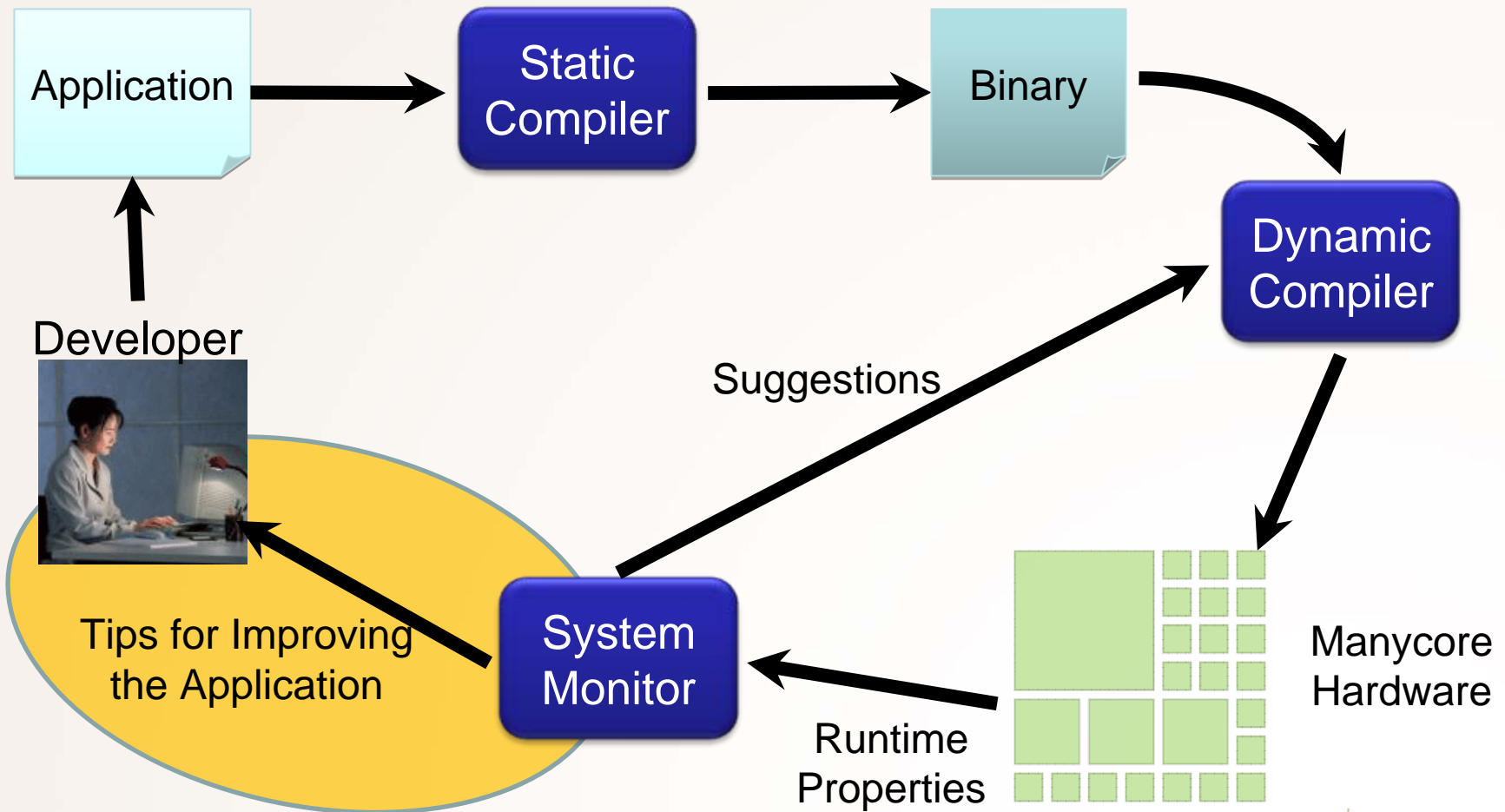
# How To Identify Data Structures

- Monitor memory allocation/accesses
  - Identify invariants on that graph



- Open questions
  - Modeling (again)
  - Other high-level properties to identify?

# Putting It All Together





# Summary

- Efficient software drives computing
  - Need to leverage parallelism/heterogeneity
- **Productivity aids!**
- Dynamic adaptation
- Synthesizing high-level semantics