



CSCI-UA.0480-003
Parallel Computing

Lecture 23: The Future

Mohamed Zahran (aka Z)

mzahran@cs.nyu.edu

<http://www.mzahran.com>



How Will the Future Look Like?

- "I think there is a world market for maybe five computers."
 - Thomas Watson, chairman of IBM, 1949
- "There is no reason in the world anyone would want a computer in their home. No reason."
 - Ken Olsen, Chairman, DEC, 1977
- "640K of RAM ought to be enough for anybody."
 - Bill Gates, 1981

Predicting the Future is not easy!!

How Will the Future Look Like?

- **Evolution:** just interpolating the current trend
- **Revolution:** a new technology, a paradigm shift → very hard to predict!

The Software

Evolution

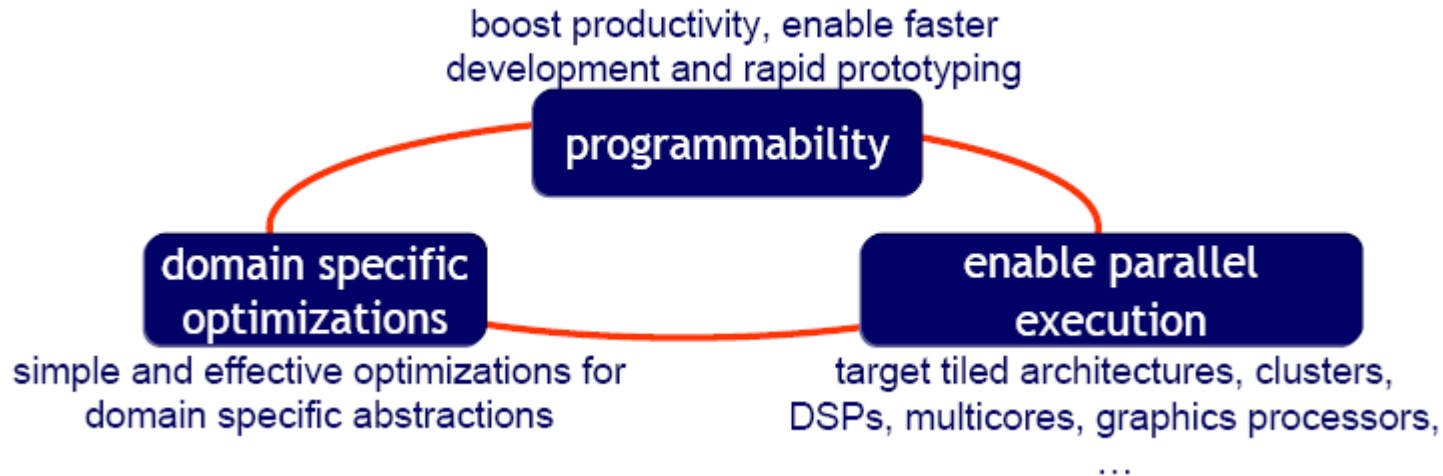
- Application types
- Languages
- MPI for multicore?
- Auto-parallelization

Example of Evolving Applications

- Content-based image retrieval
- Health-record management
- NLP
- Teleconferences
- Heavy multimedia contents
- More realistic graphics and user interface
- Event-driven (real-time)
- ...

Evolving Languages

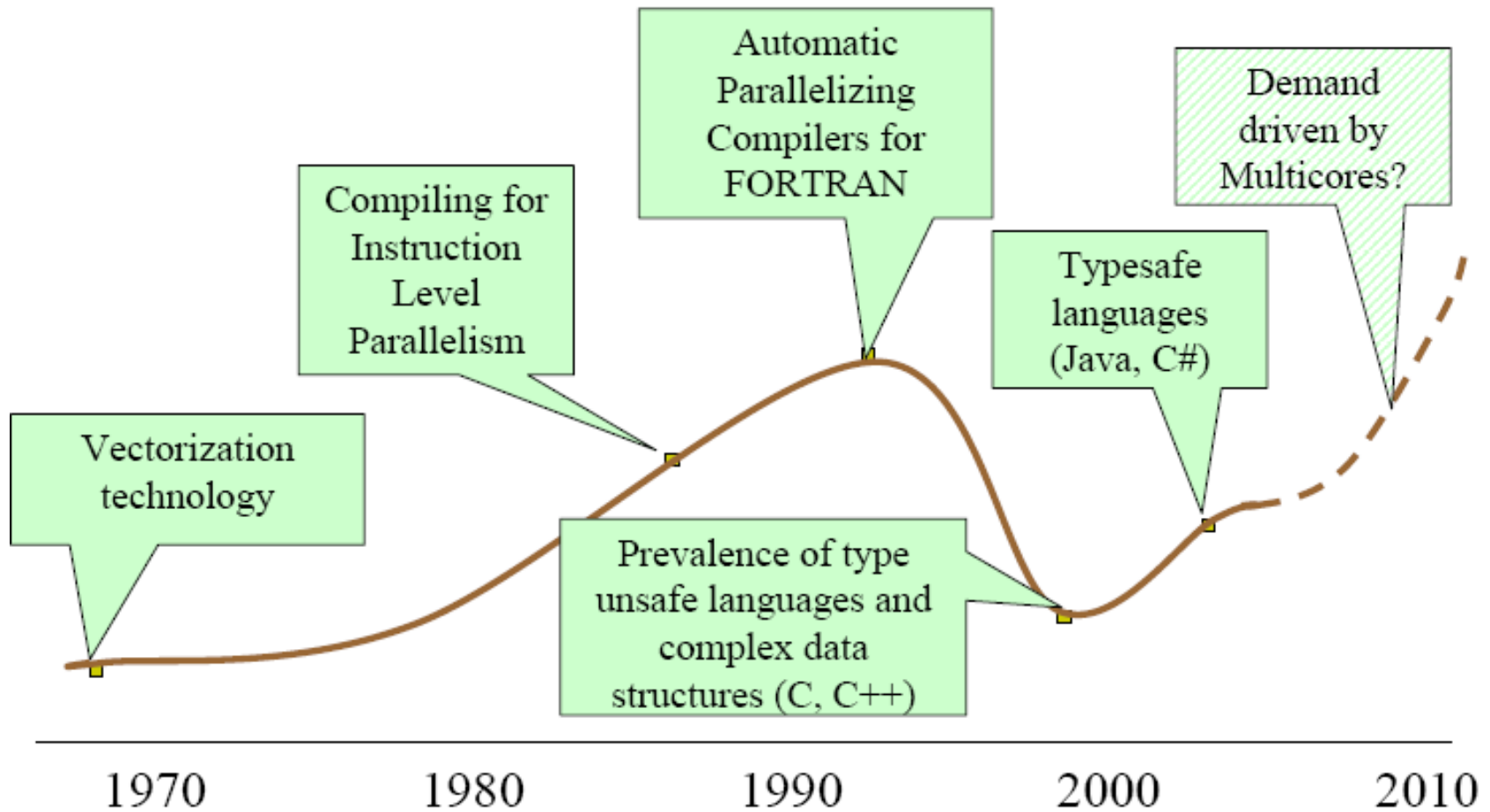
- Scripting
- Domain specific languages (e.g. StreamIt, ...)



MPI on Multicore

- One MPI process per core
 - Each MPI process is a single thread
- One MPI process per node
 - MPI processes are multithreaded
 - One thread per core
 - aka Hybrid model

Improvement in Automatic Parallelization



Source: Saman Amarasinghe, MIT

What Do We Need From the Compiler?

- Compilers are critical in reducing the burden on programmers
 - Identification of data parallel loops can be easily automated, but many current systems require the programmer to do it.
- Reviving the push for automatic parallelization
 - Best case: totally automated parallelization hidden from the user
 - Worst case: simplify the task of the programmer

How Can the Compiler Parallelize a Program?

- Find the dependency in the program
- Try to avoid or eliminate the dependency
- Reduce overhead cost

The Future of Software: Revolution

- Program design methodology
 - Sketching? → given a specification, synthesize a program meeting this spec



sketch



program = completed sketch

- New programming paradigm

The Future of Software: Revolution

- Self-Modifying code ... Again!

The Hardware

The Future of Hardware: Evolution

- More cores on-chip but constraints increase
 - Dark-silicon becomes more substantial
 - Wire delay
 - Power
 - Reliability
- More widespread heterogeneous multicore

The Future of Hardware: Revolution

- Brain-inspired machines
- Reconfigurable processors
- DNA computing
- Quantum computing
- Non Von-Neumann model
- ...

Conclusions

- Increasing the number of on-chip cores leads to many challenges both in software and hardware.
- Do we really need to increase the number of on-chip cores?
- What if we can no longer increase the number of on-chip cores?