

RunParallel

An auto-parallelizing runtime
for your code

TVP LLC

Russia

Compute architectures

Heterogeneous Clusters



x86

Multicore CPUs



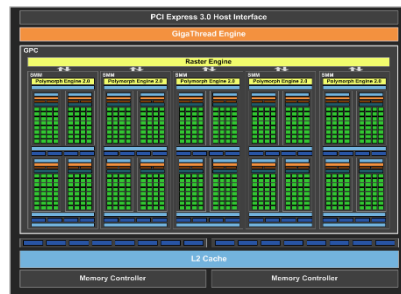
Accelerators

GPU

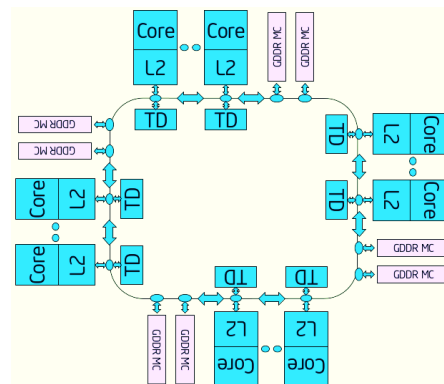
AMD GCN



NVidia Maxwell



Intel MIC



An Introduction

- What is it?
 - An auto-parallelizing runtime for your code written in Java.
- What is the purpose of the project?
 - Provide a unified technology to program all the variety of modern HPC-systems.
- Why this tool?
 - New methods to compute better space-time mappings during automatic parallelization.
 - Just-In-Time parallelization: your implementation is always cross-platform.
 - Heterogeneous clusters with accelerators are supported.

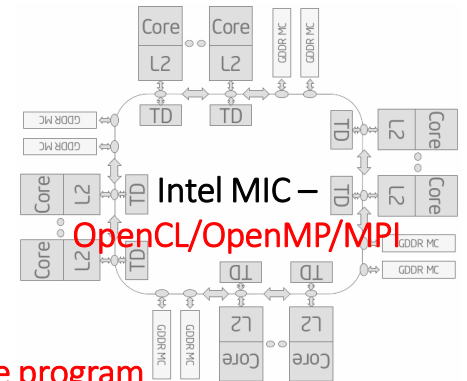
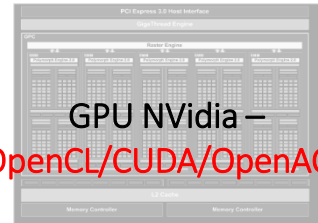
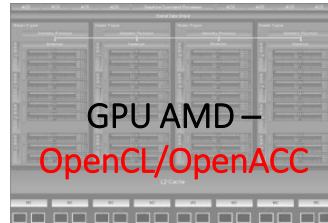
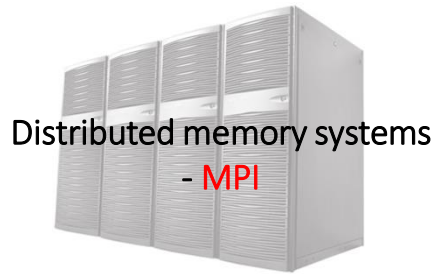
Existing parallelizing systems

Tool	Active development	Scheduling and placement optimization for data and computations	State-of-art polyhedron model methods	SAAS	Price	Supported architectures
OPC	Yes	Yes	No	No	0	x86-clusters, GPU NVidia
DVM	Yes	No	No	No	0	x86-clusters, GPU
LooPo	No	Yes	No	No	0	x86-clusters
Pluto	Yes	Yes	Yes	No	0	x86-servers, GPU NVidia
PIPS	Yes	Yes	Yes	No	0	x86-clusters, GPU
PoCC	No	Yes	Uses Pluto			
Polly	Yes	Yes	Pluto + LooPo	No	0	LLVM
Cetus	Yes	No	No	No	0	x86-servers
PGI	Yes	Yes	Unknown	No	>\$900/year	x86-clusters, GPU
CAPS	No	Yes	No	No	Bankrupt	x86-clusters, GPU
Intel Parallel Studio	Yes	No	No	No	New: \$1,449/year Renewal: \$499/year	x86-clusters, Intel Xeon PHI
RunParallel	Yes	Yes	Yes	Yes	One-time \$100 Subscription \$400/6 months	Heterogeneous clusters, x86, SMP, NUMA, NVidia GPU, AMD GPU, Intel Xeon PHI

Features and disadvantages of the analogs

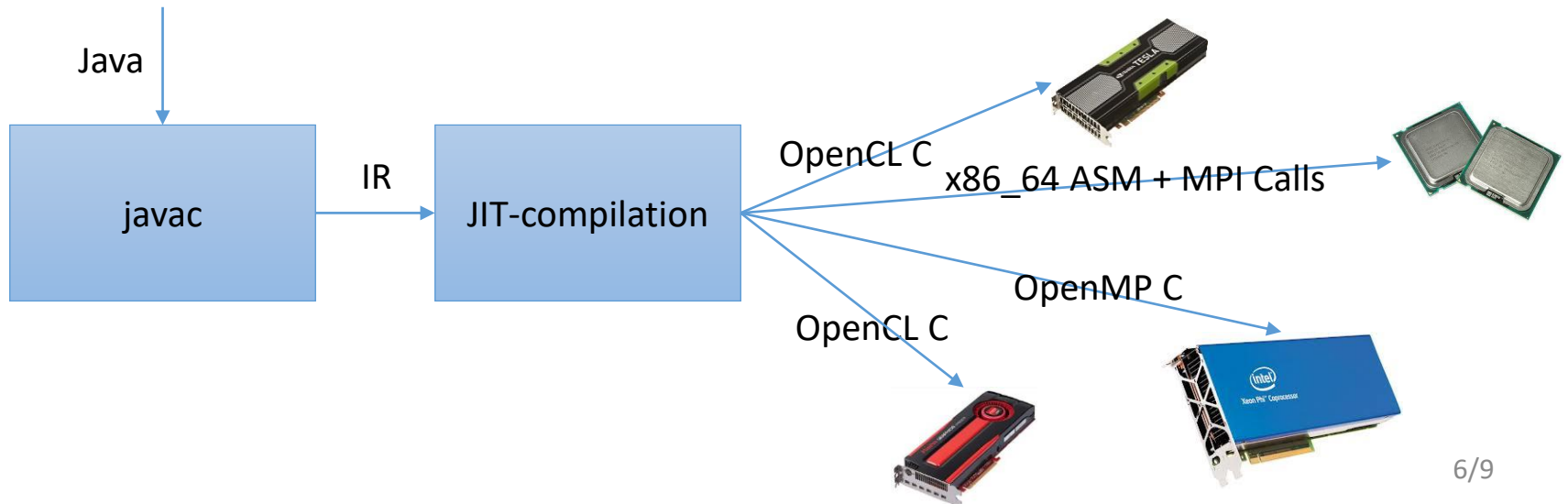
- Polyhedron model is the common mathematical background for all the free compilers and tools.
- All the compilers do static compilation only.
 - A binary image is not portable.
 - Possible obstacles to the use of polyhedron model:
 - Not all the parameters may be defined in compile-time.
 - Non-affine array indices.
- None of the compiler does cover all the variety of modern computing hardware – accelerators and systems with shared and distributed memory.

Programming HPC-systems today and tomorrow

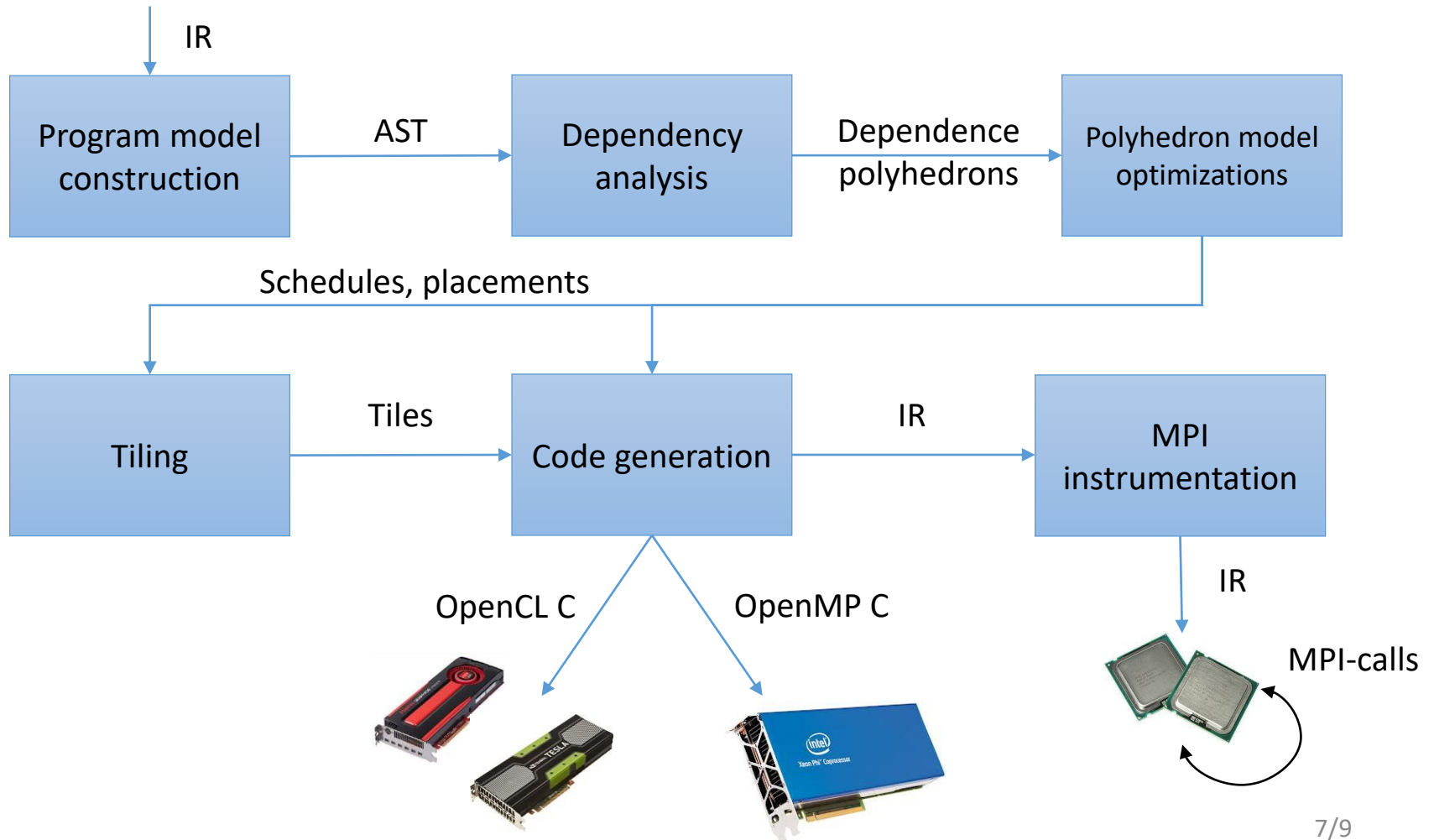


Today – a bunch of technologies, compilers, languages, implementations of the single program

Tomorrow – a unified runtime, a single language, a single implementation of the program



Parallelization phases within JIT-compilation mechanism



Efficiency and performance background

- New cost model to choose better schedules and placements for computations during automatic parallelizing. Subjective preferences are defined by weighting coefficients in linear form of scalarized multi-objective problem. This allows to specify subjective preferences more precisely in contradistinction to classic methods relying on lexicographic optimization
- New methods to compute space-time mappings intended for reducing data reuse distance in time and space domains.
 - for SMP and multicore CPUs;
 - for NUMA and clusters.
- Just-In-Time parallelization of non-affine programs.

Want to try? Contact us:

<mailto:support@runparallel.net>