MANCHESTER
1824
The University of Manchester

Enabling Virtual Machine Research on ARM-based IoT devices

Christos Kotselidis
Lecturer
APT Group
The University of Manchester

ARM Research Summit, 17-19 September 2018, Cambridge, UK

Outline

- Motivation
- Application virtualisation on ARM IoT
  - Opportunities
  - Challenges
- MaxineVM on ARM
- Conclusions

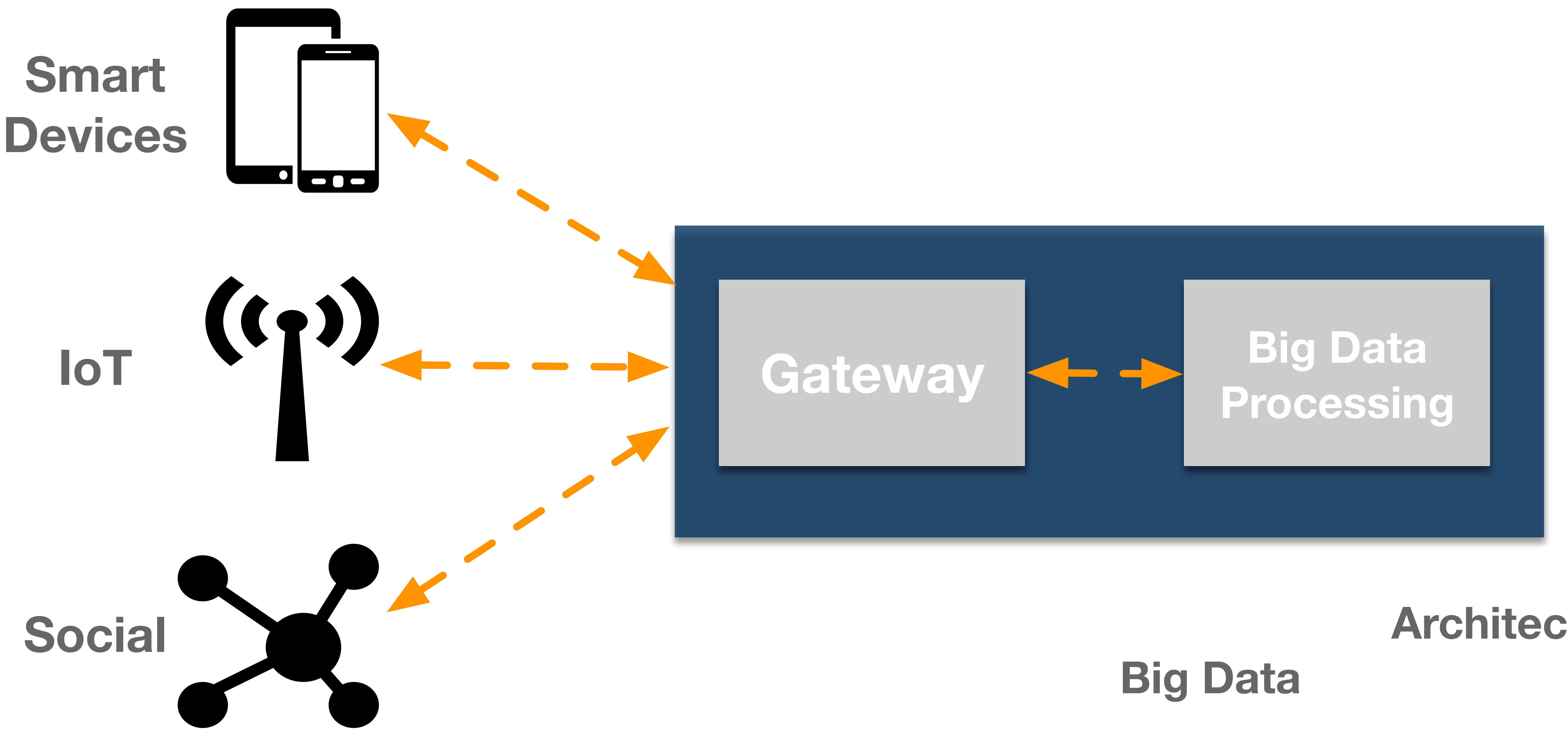
Motivation | Big Data/IoT synergy

- Big Data processing of diverse data streams
- IoT: key enabler for data generation
- Big Data frameworks consume IoT generated data
  - Directly or indirectly
- Ideally, synergistic operation for:
  - Higher performance: Throughput, Latency, etc.
  - Data throttling

Motivation | Typical deployment

→ Input Streams to the Big Data Platform

← Lightweight control protocol (e.g. MQTT): Suscsribe, Start, Stop, etc.



Architecture and Programmability

Big Data

Spark
Apache Flink
Hadoop

Scale out/up
Typically x86
Cloud/Cluster
JVM based
Rich libraries
Evolving ecosystems

IoT

ARM
Devices
Sensors
Smartphones
Honeypots

mWatts to 1-2 Watts
Resource constrained
Diversity
C, C++, Java
Micropython

Motivation | Challenges

- Code duplication and maintenance
- Security and code validation
- Lack of dynamic data control
- Inability to perform performance/energy decisions

Motivation | Application Virtualization

- Embedded JVMs on IoT devices
  - e.g. JamaicaVM, MicroEJ, IBM J9
- *Write-once-run-everywhere*
- Operator market places | Security, Verification
- Programming framework unification
  - Drag and drop operators from Big Data frameworks
  - Dynamic data-flow control

Challenges | Application Virtualisation

Inherent complexities of JVMs

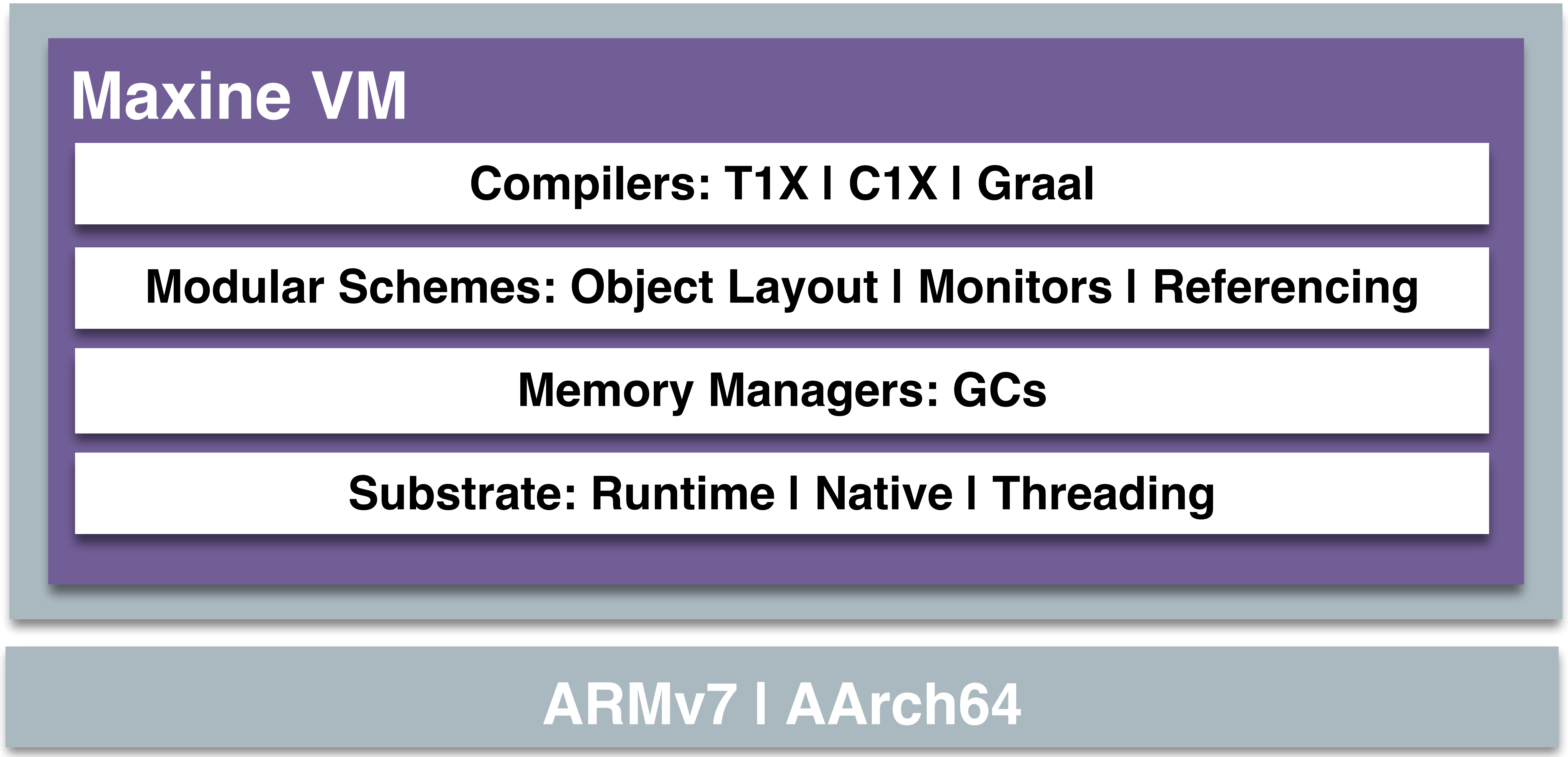
- JIT compilation
- GC overheads
- Space and memory constraints
- Real-time JVMs (RTSJ) may constrain the programming model
- Power hungry

Challenges | Research VMs on ARM

- Create a state-of-the-art Research JVM for:
  - ARMv7 | 32 bit
  - AArch64 | 64 bit
- Enable fast experimentation and prototyping
- Tackle performance/energy tradeoffs
- Perform ARM-specific optimisations

MaxineVM I Objectives and Strengths

- Highly modular
- Easy experimentation
- Scalable (32 | 64 bits)
- Comparative performance with industrial VMs
- Open source



Maxine VM

Compilers: T1X I C1X I Graal

Modular Schemes: Object Layout I Monitors I Referencing

Memory Managers: GCs

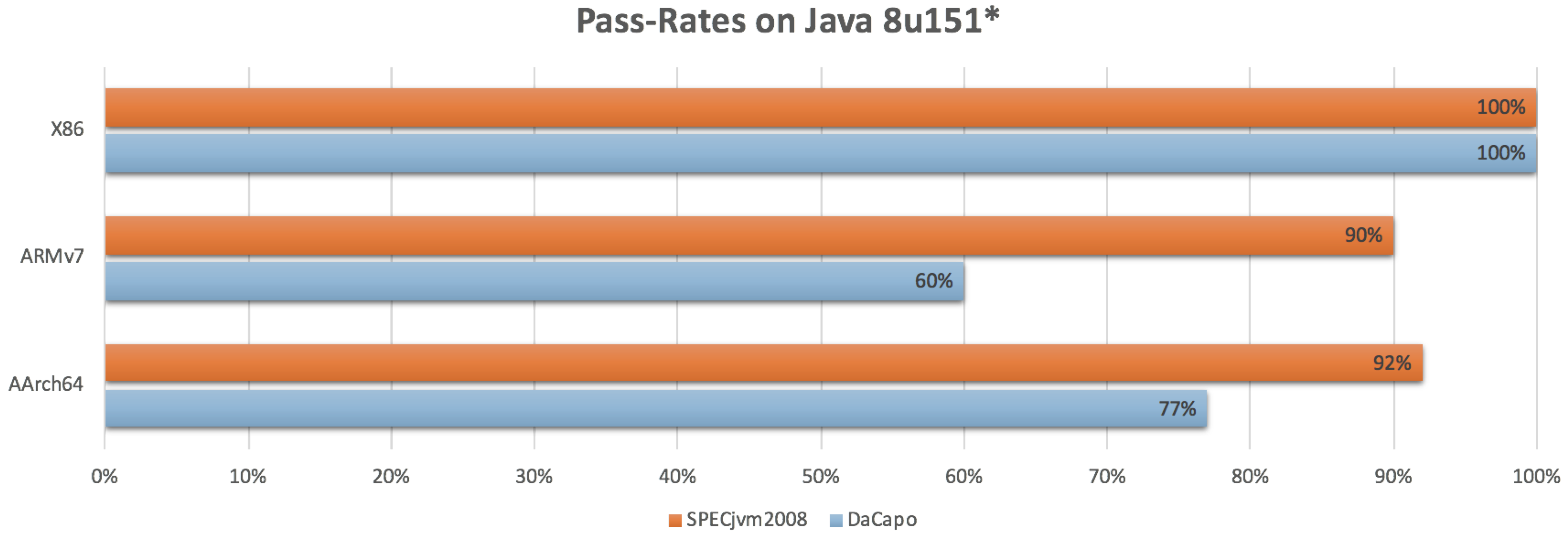
Substrate: Runtime I Native I Threading

ARMv7 I AArch64

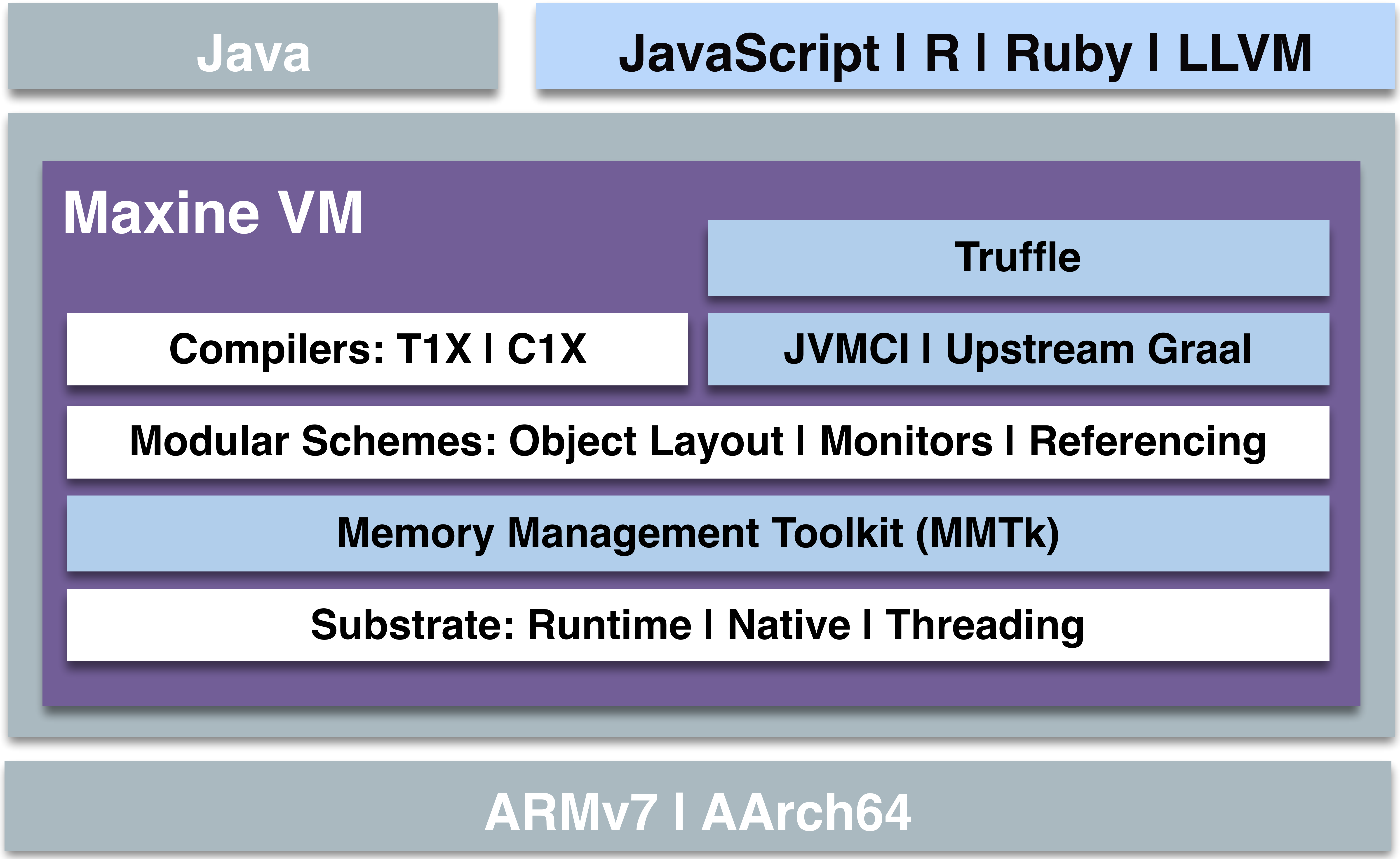
MaxineVM | Support for ARM arch

Significant engineering effort:

- Port and validation of all compilers to ARM ISA
- Increased pass-rates in standard benchmarks
- Continuous performance improvements



MaxineVM | Beehive Vision



Java

JavaScript | R | Ruby | LLVM

Maxine VM

Truffle

Compilers: T1X | C1X

JVMCI | Upstream Graal

Modular Schemes: Object Layout | Monitors | Referencing

Memory Management Toolkit (MMTk)

Substrate: Runtime | Native | Threading

ARMv7 | AArch64

Conclusions

- Benefits and challenges of application virtualisation on IoT devices
- Synergistic execution of Big Data and IoT
- Research VMs on ARM devices
- Maxine VM SOA Research VM for ARM
  - Open source: https://github.com/beehive-lab/Maxine-VM
  - Continuous development and improvements
  - Beehive: Long-term vision for full stack hw/sw co-design

Resources

- MaxineVM Documentation
  https://maxine-vm.readthedocs.io/en/latest/
- MaxineVM repository
  https://github.com/beehive-lab/Maxine-VM
- Heterogeneous Managed Runtime Systems: A Computer Vision Case Study (Paper)
  https://doi.org/10.1145/3050748.3050764
- Maxine: An approachable virtual machine for, and in, java (Paper)
  https://doi.org/10.1145/2400682.2400689
- Cross-ISA debugging in meta-circular VMs (Paper)
  https://doi.org/10.1145/3141871.3141872
- Beehive: A Hardware/Software Co-designed Stack for Runtime and Architectural Research (Vision Paper)
  https://arxiv.org/abs/1509.04085

Acknowledgments

- EPSRC grants
  - Anyscale EP/L000725/1
  - PAMELA EP/K008730/1
  - DOME EP/J016330/1
- EU H2020 ACTiCLOUD 732366
- ARM iCase

Andrey Rodchenko
Jiaqi Liu
Yaman Cakmakci
Iain Apreotesei
Andreas Andronikakis
Orion Papadakis
Tim Hartley
Benjamin Bell
Andy Nisbet
Foivos Zakkak
Mikel Lujan
Christos Kotselidis

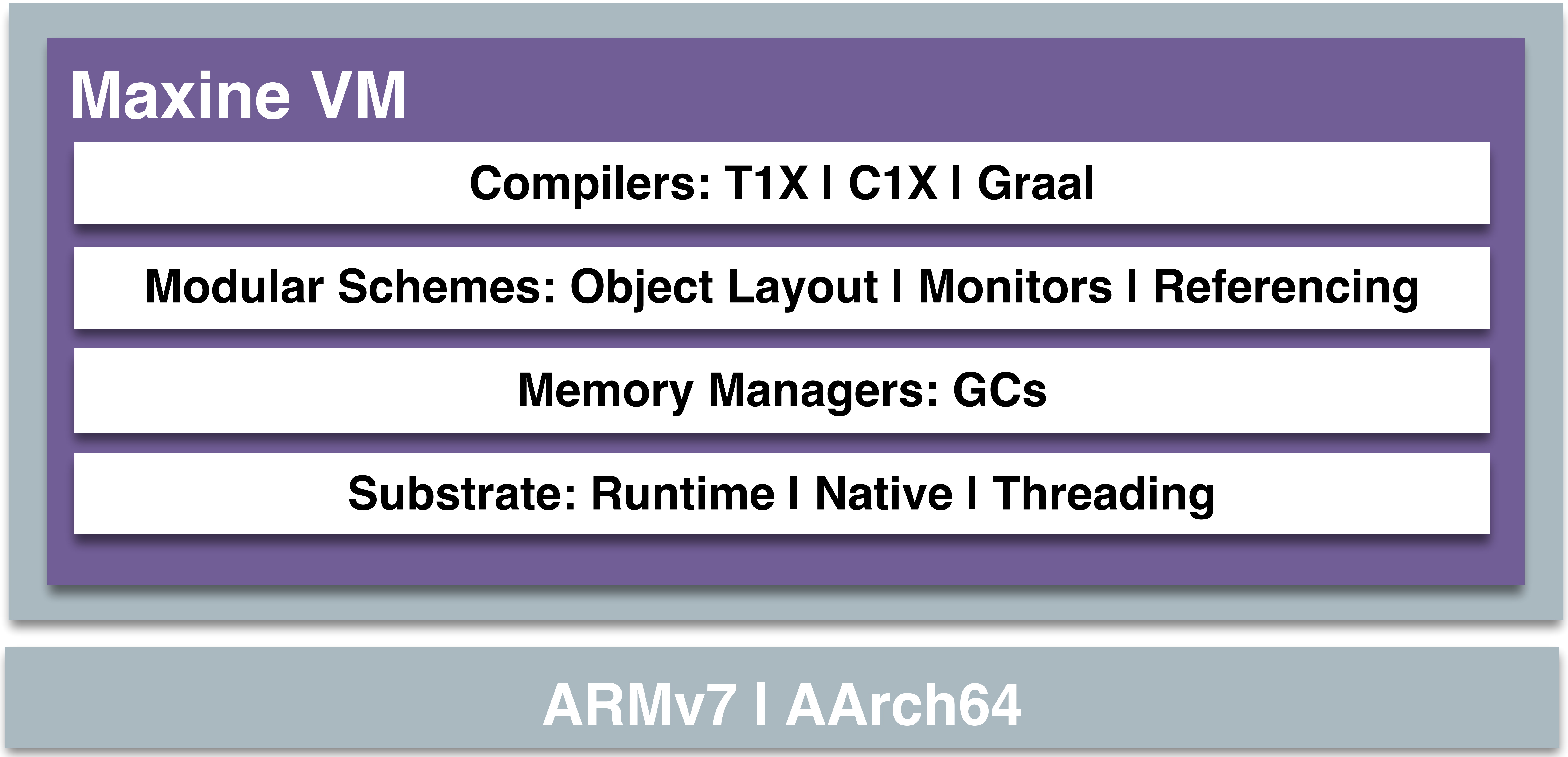
Thank you!

Questions?

https://github.com/beehive-lab

Backup slides

MaxineVM on ARM I Overview



Maxine VM

Compilers: T1X I C1X I Graal

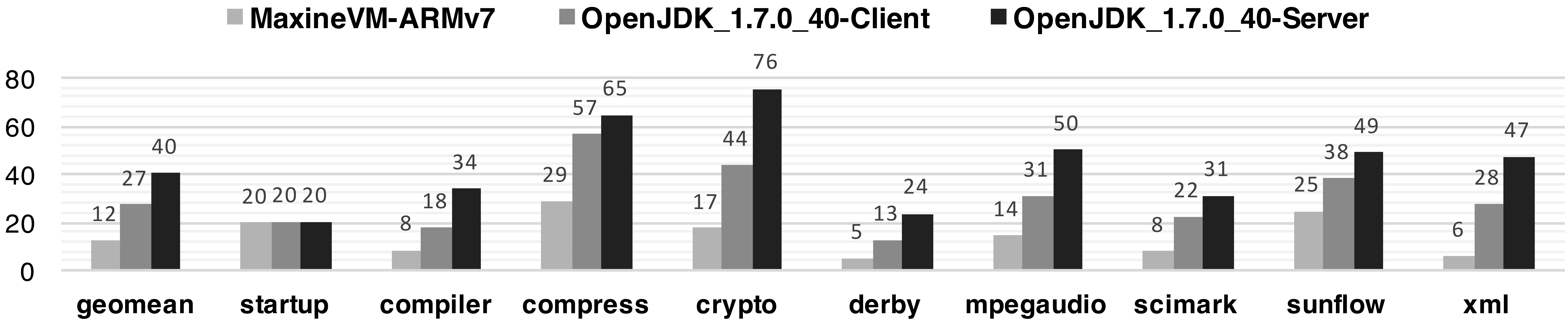
Modular Schemes: Object Layout I Monitors I Referencing

Memory Managers: GCs

Substrate: Runtime I Native I Threading

ARMv7 I AArch64

MaxineVM | Pushing performance boundaries


MaxineVM-ARMv7
OpenJDK_1.7.0_40-Client
OpenJDK_1.7.0_40-Server
geomean 12 27 40
startup 20 20 20
compiler 8 18 34
compress 29 57 65
crypto 17 44 76
derby 5 13 24
mpegaudio 14 31 50
scimark 8 22 31
sunflow 25 38 49
xml 6 28 47