CloneCloud – Elastic Execution between Mobile Device and Cloud

Byung-Gon Chun
Intel Labs Berkeley

Sunghwan Ihm
Princeton University

Petros Maniatis
Intel Labs Berkeley

Mayur Naik
Intel Labs Berkeley

Ashwin Patti
Intel Labs Berkeley

Presenter : Das

Motivation

- With the increasing use of mobile devices, mobile applications with richer functionalities are becoming ubiquitous.
- But mobile phone devices are limited by their resources for computing and power consumption.
- Cloud – the place for abundant resources
- Clouds provide opportunity to do huge computations quickly and accurately

Common approach

- Split in traditional client-server paradigm, pushing most computation to the remote server

- Few approaches
  - Cloudlets
  - MAUI

Questions

- Will it be efficient to move all the computations to cloud?
- If programmer has to specify which computation to be run in cloud, will that be same for all mobile configuration like Faster CPU with good network connection, Slower CPU with slow network connection.

Goals

Main goal

"….a flexible architecture for the seamless use of ambient computation to augment mobile device applications and energy-efficient"

Other aims:

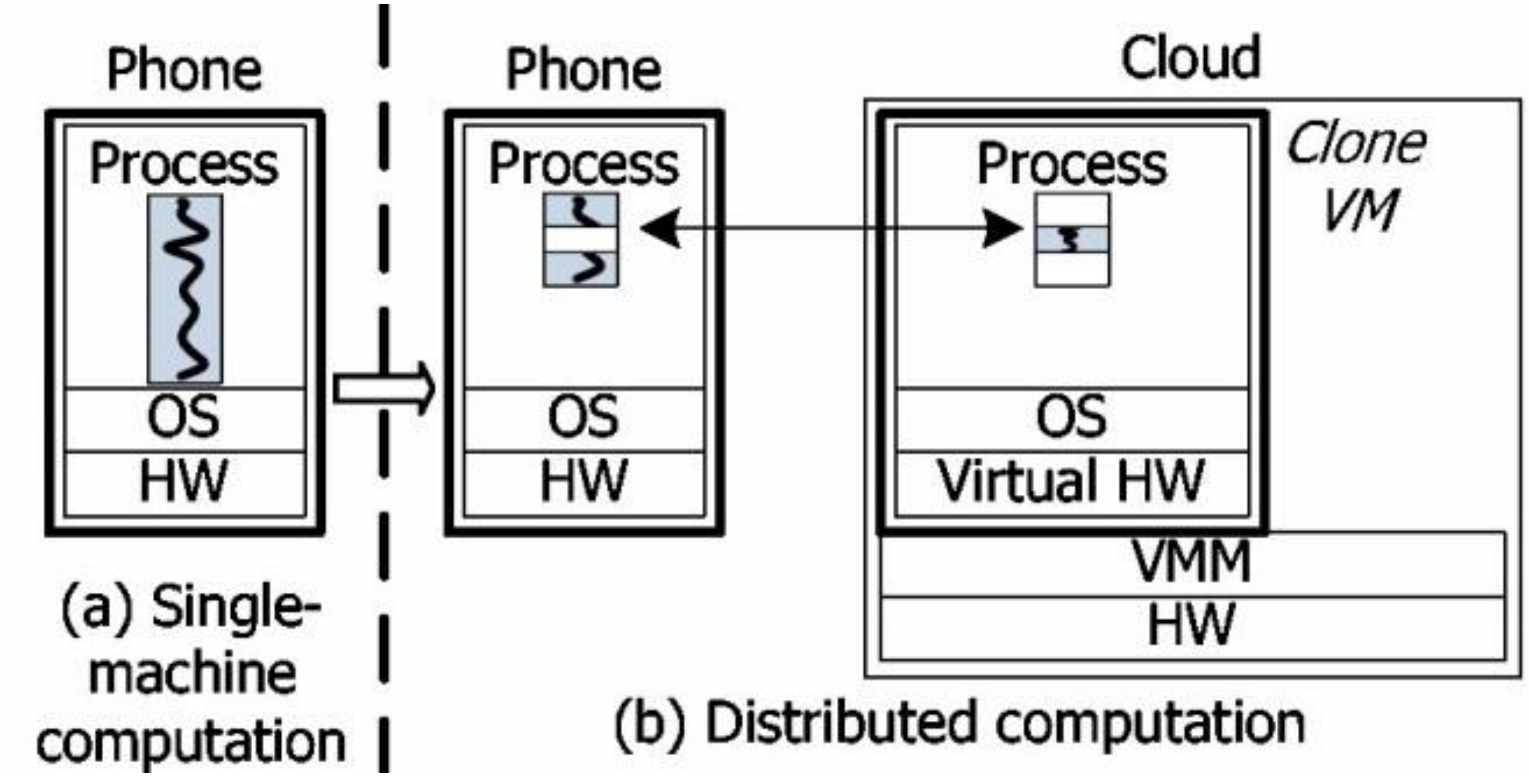
- Allow such fine grained flexibility on what run where(on mobile or in cloud)
- Take programmer out of the business of application partitioning(make it automatic and seamless)

CloneCloud

- CloneCloud envisions at an architecture that uses cloud to do computations that consume resources badly on mobiles.
- It believes in the intuition that "as long as execution on the cloud is significantly faster than execution on the mobile device, paying the cost for sending the relevant data and code from device to the cloud and back may be worth it"
- It aims at finding the right spots in an application automatically where the execution can be partitioned and migrated to the cloud.

Schema

- Clone an unmodified application executable.
- The modified executable is running at mobile device.
- At automatically chosen points individual threads migrate from mobile device to device clone in a cloud.
- Remaining functionality on the mobile device keep executing.
- Remaining functionality blocks if attempts to access migrated state.
- The migrated thread executes on the clone, possibly accessing native features on hosting platforms(fast CPU, hardware accelerations)
- Merge remote and local state back into original process.


Phone
Process
OS
HW
Phone
Process
OS
HW
Cloud
Process
Clone VM
OS
Virtual HW
VMM
HW
(a) Single-machine computation
(b) Distributed computation

Partitioning

- Partitioning mechanism yields the partitions in the application that are optimal at execution time and energy consumption
- It is run multiple time under different conditions and objective functions – stores all partitions in a database.
- It is done offline
- At run time, the execution picks a partition among these modifies the executables before invocation
- It has three components – static analyzer, dynamic profiler and optimization solver.

Partitioning

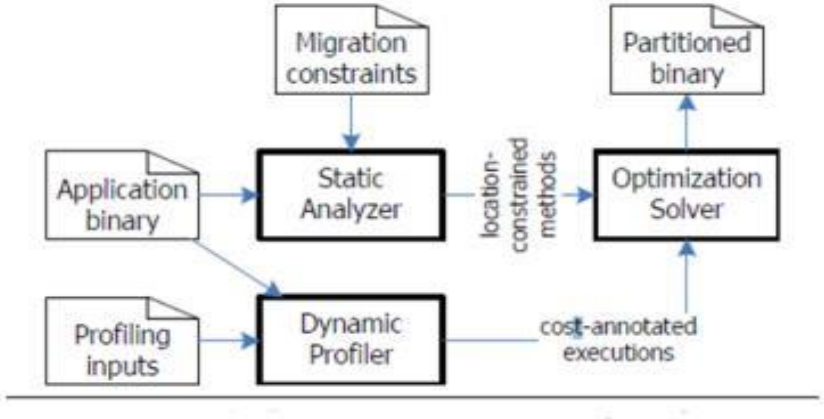


Fig 5: Partitioning Analysis Framework

Static Analyzer

- The static analyzer identifies the legal partitions of the application executable according to the set of constraints
- Migration is restricted to the method entry and exit points
- Two more restrictions for simplicity
  - Migration is allowed only at the boundaries of application methods but not core system library methods
  - Migration is allowed at the VM-layer method boundaries but not native method boundaries.

Static Analyzer - example

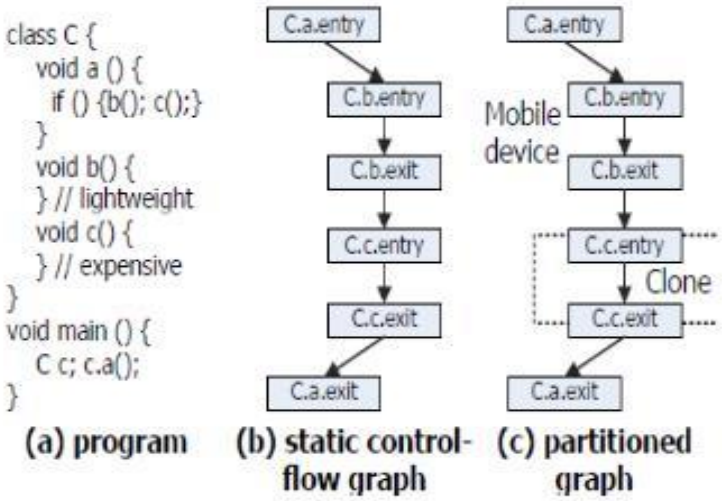


Fig 6: Flow Diagram in a Static Analyzer

Static Analyzer – constraints

Three properties of any legal partition

- Methods that access specific features of a machine must be pinned to the machine[Vm]
  - Static analysis marks the declaration of such methods with a special annotation M
  - Done once for each platform, not repeated for each application
- Methods that share native state must be collocated at the same machine[Vnat]
  - When an image processing class has initialize, detect and fetch result methods that access native state, they need to be collocated
- Prevent nested migration
  - Static analysis of the control flow graph to identity the set of methods called directly by a method(DC) and transitively(TC).

Dynamic Profiler

- Profiler collects data that will be used to construct the cost model.
- Currently using randomly chosen set of inputs
  - Future work is to explore symbolic-execution-based techniques since randomly chosen inputs may not explore all execution paths
- Each execution is run once on mobile device and once on the clone in the cloud.
- Profiler outputs set of execution S and a "profile tree", for both mobile device and the clone.

Dynamic Profiler- example

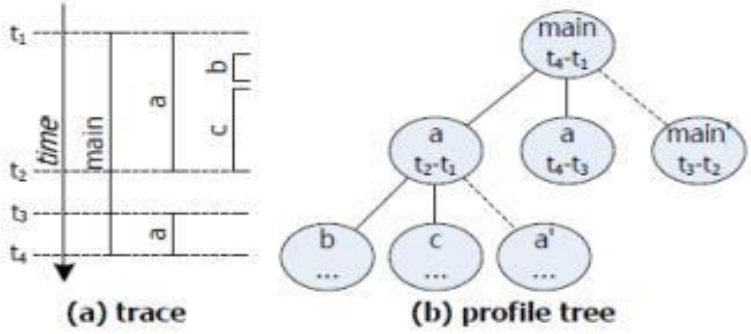


Fig 7: Example of Dynamic Profiler

Profile Tree

- One node for each method invocation
- Every non-leaf node also has a leaf child called its residual node.
- Residual node holds residual cost which represents the cost of running of the body of code excluding the costs of the methods called by it
- Each edge is annotated with the state size at the time of invocation of the child node, plus the state size at the end of that invocation
  - Amount of data that migrator needs to capture and transmit in both directions if edge were to be migration cost
- Computation cost Cc(I,l); I=0 on mobile device and filled from T, I = 1 on the clone and filled from T'
- Migration cost Cs(i); sum of a suspend/resume cost and the transfer cost.

Dynamic Profiler

- For energy consumption model, we do the energy measurements with off-board equipment.
- CPU activity(processing/idle) , display state(on/off), and network state (transmitting or receiving/idle), and translate them to a power value using func P.
- Cc(i,0) = P(CPUOn, ScrOn, NetIdle) * T[i]
- Cc(i,1) = P(CPUIdle, ScrOn, NetIdle)
- Cs(i) = <Cs(i) value from time model> * P (CPUOn, ScrOn, netOn)

Optimization Solver

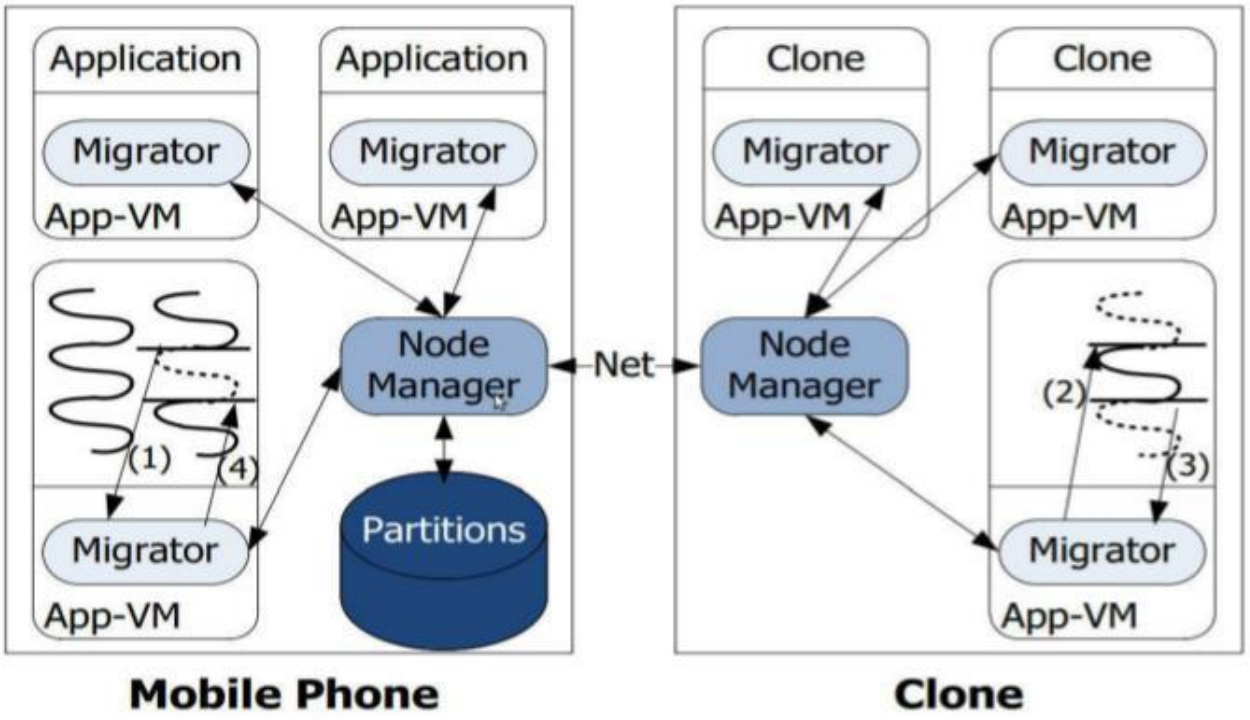
- It aims at picking up application methods to migrate the clone from mobile device, so as to minimize the expected cost of partitioned application.

- Optimization problem is to minimize the time taken to execute and energy consumed.

Distributed Execution

- Thread granularity migration
  - Migration operates at the granularity of a thread.
- Native-Everywhere:
  - Enables migrated threads to use native non-virtualized hardware (GPUs, Cryptographic accelerators, etc).

Migration Overview



- **Suspend and capture**
  - Thread migrator suspends migrant thread
  - Captures its state, passes it to node manager
  - Node manager transfers the capture to clone

- **Resume and Merge**
  - Clone's thread migrator captures and packages the thread state
  - Node manager transfers the capture back to the mobile device
  - Migrator in the original process is given the capture for resumption

Evaluation

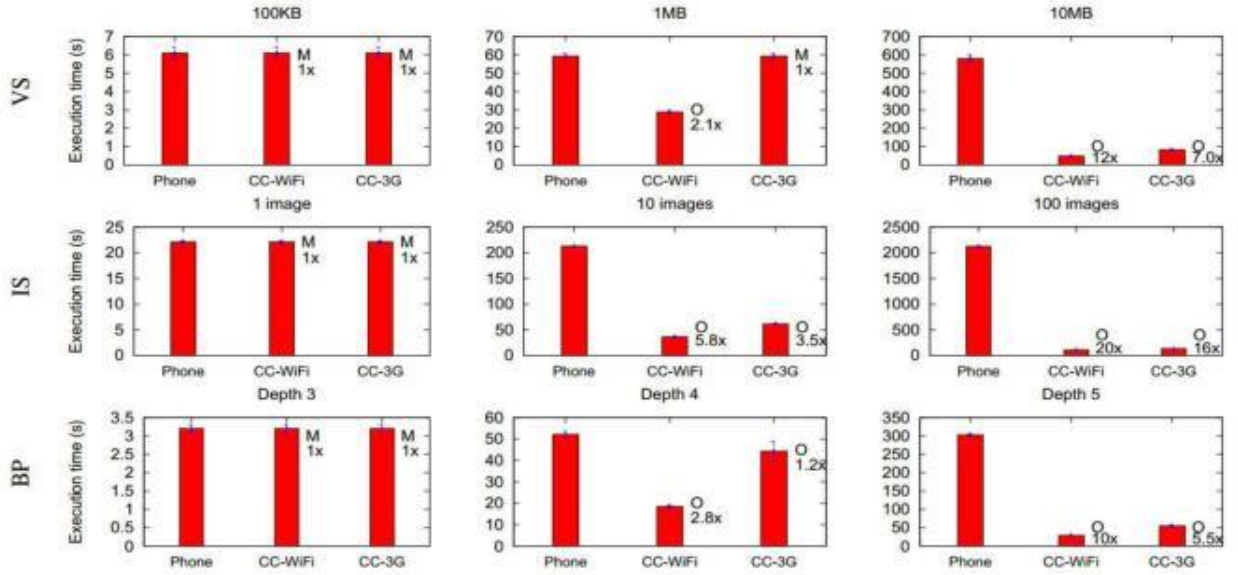


Figure 8. Mean execution times of virus scanning (VS), image search (IS), and behavior profiling (BP) applications with standard deviation error bars, three input sizes for each. For each application and input size, the data shown include execution time at the phone alone, that of CloneCloud with WiFi (CC-WiFi), and that of CloneCloud with 3G (CC-3G). The partition choice is annotated with M for "monolithic" and O for "off-loaded," also indicating the relative improvement from the phone-alone execution.

Evaluation

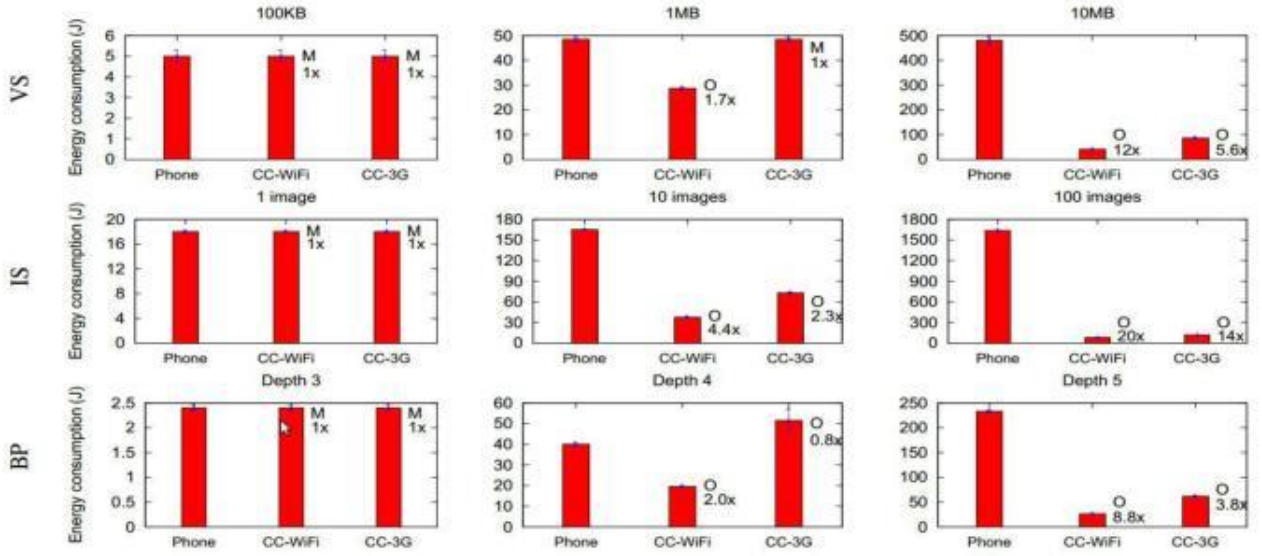


Figure 9. Mean phone energy consumption of virus scanning (VS), image search (IS), and behavior profiling (BP) applications with standard deviation error bars, three input sizes for each. For each application and input size, the data shown include execution time at the phone alone, that of CloneCloud with WiFi (CC-WiFi), and that of CloneCloud with 3G (CC-3G). The partition choice is annotated with M for "monolithic" and O for "off-loaded," also indicating relative improvement over phone-only execution.

Conclusion

- Pros
  - Prototype delivers up to 20x speed up and 20x energy reduction.
  - Programmer involvement is not required.
- Cons
  - Programmer does not have the flexibility to specify a method to be executed on cloud.
  - It may not be possible to run dynamic profiler for all possible parameters.

Questions?