Achieving Ubiquity through Hardware Virtualization

Mahadev Satyanarayanan, Carnegie Mellon University

1 Vision: *in vivo* Pervasive Computing

Mark Weiser's vision has been characterized as *"the creation of environments saturated with computing and communication capability, yet gracefully integrated with human users."* [18]. Today, we are indeed "saturated with computing and communication capability." The decade 2001-2011 saw impressive advances in the functionality, performance, and cost of mobile computers, flat-panel displays, sensors, and wireless communication — today's smartphones bear witness to these advances. However, "gracefully integrated with human users" still remains elusive.

Attaining this goal will require systems to conserve *human attention*, the most precious resource of all [21, 22]. This requires elimination of system-induced distractions such as failures, poor or erratic performance, confusing output, and out-of-context interactions. By trading off plentiful computing resources for this scarcest of resources, the end-to-end effectiveness of systems in human workflows can be greatly improved. Only through this path can pervasive computing approach Weiser's eloquent ideal of a disappearing technology: *"They weave themselves into the fabric of everyday life until they are indistinguishable from it."* [25]

Researchers worldwide have been exploring diverse approaches towards this overarching goal. The next decade (2011-2020) will be one of synthesis and integration, where individual mechanisms for lowering distraction under specific conditions (such as context awareness, location transparency, disconnected and weakly connected operation, application-aware adaptation, cyber foraging, and so on) are transformed into a coherent whole. Inevitably, the individual modules of such systems will have high *external complexity,* with wide programming interfaces.

Naive deployments of these complex systems at scale in the real world will fare miserably. A foretaste of our likely experience can be inferred from our collective experience with *process migration.* The earliest demonstrations of this mechanism date back to the early 1980s [16, 24]. Every five years or so, there is renewed interest in some flavor of this mechanism [2, 8, 15, 26]. Yet, even today, there is no production-quality, ubiquitous deployment of this mechanism. The reason is because it is a *brittle abstraction.* A typical implementation of process migration involves so many external interfaces that it is easily rendered incompatible by a modest external change. Long-term maintenance of software infrastructure with support for process migration involves too much effort relative to the benefits it provides. Unless explicitly addressed, the brittleness of complex software systems will plague the grand unified pervasive computing systems of the next decade.

Hardware *virtual machine (VM)* technology can rescue us here. A VM transforms external complexity into internal complexity. Contrast today's deployments of VM migration in cloud computing to the wallflower-like status of process migration, in spite of its considerable efficiency advantages. A VM offers clean encapsulation and separation: the interface between the host and guest environments is narrow, stable, and ubiquitous. Malleable software interfaces and fragile interconnections are hidden inside the guest environment and transported in their entirety. By lowering the external complexity of a software system, a VM greatly simplifies its deployment. The next section presents early evidence of the relevance of VMs to pervasive computing.

2 Evidence: VM-based Cloudlets and TransientPCs

Cyber foraging, which is the use of proximate hardware infrastructure to augment the computing, storage, and energy resources of mobile hardware is now widely recognized as a critical enabling technology in pervasive computing [3, 4, 5, 9, 10, 11, 13, 14, 18, 23]. "Find hardware nearby and use it" sounds simple. However, this conceptual simplicity masks deep practical challenges. At their heart is an inherent tension between the viewpoints of users and infrastructure owners. A user wants effortless, safe, fast and ubiquitous access to nearby infrastructure that is precisely configured for his use, regardless of where he is in the world. The owner of infrastructure seeks adequate return on his investment, which translates to effortless setup and minimal management combined with the ability to serve the widest possible range of users while remaining safe from careless and malicious users. Unless both viewpoints can be simultaneously satisfied, cyber foraging will never be deployable at scale and will thus have little real-world impact. We have recently described how the use of VMs can make this problem tractable [19]. Briefly, the idea is to deploy generic *cloudlets* on which customized VMs can be dynamically instantiated by each mobile device. The guest environment of such a VM is customized software that is tailored to exactly match an application on the mobile device. Thus, wherever a mobile device goes in the world, it temporarily customizes its neighborhood to precisely match its requirements. The use of VMs avoids cloudlet customization conflicts when multiple mobile devices are in the same neighborhood. While this concept is still at an early stage of investigation, it offers high potential for deployments of pervasive computing.

VMs are also central to a broad class of systems that we refer to as *TransientPC systems.* The Internet Suspend/Resume system [12, 20], SoulPad [6], the Collective [17, 7], and Moka5 [1] are examples of this class of systems. While they differ considerably in their technical details, all these systems share the top-level goal of decoupling personal computing (PC) state from hardware and using VMs to encapsulate and precisely re-create that state anywhere and at any time. Even as pervasive computing gains momentum, the large installed based of PC software in the world will not disappear abruptly. We envision an extended period, lasting many years, when "old world" PC software continues to coexist with "new world" pervasive computing software. Each of the major inflection points in computing history (batch to timesharing, timesharing to personal computing) has required an extended period of transition. VM-based TransientPC systems can help to ensure that the next transition, from personal to pervasive computing, is graceful and non-disruptive.

3 Background and Experience of Participant

I have been working in the broad area of mobile and pervasive computing since its earliest days (circa 1990), and have originated many of the seminal concepts in the area. Details of my research contributions can be found in my bio at `http://www.cs.cmu.edu/~satya`. My former PhD students and their students now contribute extensively to mobile and pervasive computing research at a number of universities and research labs.

References

One page of References omitted since EDAS refuses to allow upload. I can either (a) delete all citations, or (b) you can give me one extra page just for the references.