Application and Infrastructure Challenges in Pervasive Computing

Swarup Acharya
Bell Laboratories, Lucent Technologies, Inc.
acharya@bell-labs.com

1 Introduction

The technological advances over the last few years has revolutionized the world of personal computing. Today, handheld devices exist that can function as a cellular phone, video camera and a PDA, all in one box. Also, wireless service providers have started offering numerous data services over their networks (e.g., NTT DoCoMo's I-Mode). With all these diverse technologies coming together, the vision of *pervasive computing* — anywhere, anytime data access on any device is finally beginning to take shape[1].

In order to migrate from a pervasive computing vision (or, hype?) into reality, fundamental issues in numerous areas — networking, data management and security among others, have to be addressed. *The need above all is that of "killer apps"* — applications with mass appeal that are economically attractive. Vertical domain specific applications exist today (e.g., mobile inventory tracking used by UPS) but the growth of the field is limited unless applications move beyond niches. Today, handheld and mobile devices are abound; the leap will occur when these devices can be networked together into a creative, useful application with wide public appeal.

In this paper, we thus, take an application-centric view of the challenges, namely, what spaces are ripe for killer apps and the appropriate data management infrastructure needed to support them. While a brand new killer app is always a possibility, we focus on how existing applications might adapt and be effective in this environment. To do that, one needs to first understand the unique environmental constraints. From a data management perspective, the following three key issues need to be noted:

- **Resource Impedance Mismatch:** Even though handheld and mobile devices have come a long way, they are still a generation away in terms of capabilities compared to desktop systems. This impedance mismatch is apparent on nearly all aspects – processing power, screen size, battery power etc. Additionally, bandwidth for wireless access is clearly lower compared to wireline networks even with the arrival of next generation wireless 3G networks. In fact, one may easily argue that *pervasive systems will always be one order of magnitude or more inferior to that of traditional computing systems*.
- **Scalability:** The mobile, wireless environment is conducive for applications in the information-centric arena. Examples of such applications include wireless internet access and real-time traffic planning systems. Scalability for such applications is a serious issue on many fronts – number of users (e.g., a traffic information system can have 100,000+ users in rush hour), physical spread of the client base (e.g., continental USA) and variety of devices (PDAs, cell phones, laptops). Additionally, the application data is likely to have a real-time component requiring specific

[1] In this document, ubiquitous computing and wireless computing will be used interchangeably to also denote pervasive computing

delivery deadlines.

- **Mobility:** Mobility can arise in two forms in a pervasive network — mobility of devices (such as driving with a cell phone) and mobility of users from device to device (with an expectation of their environment following them). For data management, mobility is a less of an issue if the underlying network layer hides it. In fact, mobility opens up a slew of new applications such as location-based services.

Given these differences (and more), research can focus two ways in designing pervasive applications. One approach is to start with a clean slate and design the end-to-end system from scratch. In the paper, we explore the alternate approach — given existing applications and infrastructure setups in wired and wireless systems, how can they be best exploited to enable next generation killer apps.

2 Research Directions

2.1 Reinventing Applications

One can argue that mobile devices will always be one order of magnitude "weaker" than their desk-bound counterparts. Consequently, any application expected to run on these devices has to account for small screen sizes, low processing power, smaller storage and memory capacity and lower network bandwidth. Running a regular desktop application as-is on these devices will likely render them too unresponsive. Clearly, there is a need for alternate thinking and out-of-box research on how regular applications can be architected to provide reasonable levels of responsiveness inspite of weaker capabilities of the hardware.

One approach to address this issue is that of "approximation" wherein established notions of *precise, exact data* is replaced by fuzzy, *approximate data*. Below are two examples where such an approach have enabled wireless web access and database processing to migrate to mobile systems. These systems aim to work with a subset of the data to provide acceptable performance at the expense of loss of response "quality". As long as the response is reasonable, it makes a good performance-quality tradeoff for pervasive systems.

- **Wireless Web Access:** Given small screen sizes, Web sites have to massage data to fit in small spaces. In one approach, content providers create two distinct types of content – one for desktop boxes and another solely for handheld access which have minimal images and text (e.g., as done by Yahoo). The downside of this approach is that if a site hasn't been setup for handheld access, it is out of bounds. The other approach is to dynamically "translate" regular content on-the-fly for handheld devices. This is done by shrinking the size and "quality" of images/text with often marginal impact to the human eye due to the screen quality and size. This increases the breadth of viewable sites and avoids the hassle of two independent systems for content providers. Interested readers are referred to [AKP99, NET01] for commercial and research systems which follow this approach.
- **Small Footprint Database Systems:** To design a database system that can scale down to mobile devices, one has address both the footprint of the database and the volume of data processed. An approach proposed recently to address the latter problem is that of approximate querying systems [AGPR99] wherein the database system returns approximate answers with specified error bounds (e.g., average salary is \$35K $\pm$ 3K). These systems work of a statistical sample of the original data (1-5% of data size) and thus, require minimal storage and have very fast response times.

Clearly, approximation exploits the performance-quality tradeoff to adapt current application to this environment. Research needs to focus on other such tradeoffs that are appropriate and acceptable to the pervasive environment.

2.2 Architecting Data Servers

A natural issue for debate is the data delivery architecture that will enable applications to span both the wireline network and the mobile users. Given this dichotomy, the focus should be on a architecture that bridges the two contrasting domains. We propose a "data server" architecture model. Similar to WAP proxies for handheld devices or, the wireless base station in a cellular network, a data server is the final wired entity before the start of the "wireless" last mile. In effect in a client-server model, to the mobile client, the data server acts as a *server surrogate*.

A data server approach has has a number of advantages. It can isolate the vagaries of the wireless domain from the rest of the wired environment and similarly, tailor the data to the format accepted by the mobile device. Given that the data server is the source of data for the mobile, there are a number of opportunities for research in how data may be "tailored" and the protocols that would enable it. We highlight some of these below:

- Data delivery model appropriate for different applications– push, pull, unicast etc. ([FZ96]).
- Research on efficient scheduling algorithms (e.g., [AM98, AF98]).
- Research on communication protocols (possibly, proprietary) which will enable efficient data transfer between the data server and the mobile (e.g., end-to-end or, proxy-based networking).
- Research on services built on the top of the basic communication layer – encryption/security, transactional support, disconnected access etc.

While the data server can serve as the bridge, it has potential to become a single point of failure. Data server architectures should thus, additionally provide for fault tolerance while aiming for high performance.

3 Conclusions

Pervasive computing is on a cusp of a revolution. The combination of powerful, mobile devices and universal network connectivity is close to critical mass. The next step is the availability of killer apps that can exploit the environment to provide useful services. The key area of research focus should be on infrastructure and hooks for application designers so that these killer apps may be built and deployed.

References

[AF98] D. Aksoy and M. Franklin. Scheduling for large scale on-demand data broadcast. In *Proc. of IEEE INFOCOM*, San Francisco, CA, March 1998.

[AGPR99] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. Join synopses for approximate query answering. In *Proc. ACM SIGMOD International Conf. on Management of Data*, pages 275–286, June 1999.

[AKP99] S. Acharya, H. F. Korth, and V. Poosala. Systematic multiresolution and its application to the World W ide Web. In *Proceedings of the International Conference on Data Engineering*, Sydney, Australia, March 1999.

[AM98] S. Acharya and S. Muthukrishnan. Scheduling on-demand broadcasts: New metrics and algorithms. In *ACM Mobicom*, pages 43–54, New York, October 25–30 1998. ACM Press.

[FZ96] M. Franklin and S. Zdonik. Dissemination-based information systems. *Data Engineering Bulletin*, 19(3):20–30, 1996.

[NET01] AppCelera Burns Up the Last Mile. Network Computing Magazine Article. WWW URL: `http://www.networkcomputing.com/1224/1224f3.html`, Dec 2001.