Archer - Seeding a Community-based Distributed Computing Infrastructure for Computer Architecture Education and Research

Summary:
We are developing a distributed shared resource that will address computational needs of the computer architecture community at large. Our goal is to create a high-performance computing community resource such that any individual researcher, student, educator can easily tap into the infrastructure, and at the same time contribute with resources of their own – be it a desktop at home or work, or a lab cluster. We are seeking input and support to make this system a unique resource to the architecture research and education community, and we would like to invite you to test our prototype and provide us with feedback.

What is the current infrastructure?
We have developed and deployed a proof-of-concept prototype infrastructure and we are putting together a proposal to the community resource development track of the National Science Foundation CISE computing research infrastructure proposal (CRI) program to obtain funds to seed the deployment of Archer, a high-performance distributed infrastructure for computer architecture simulation in research and education. The prototype Archer has been deployed on tens of CPUs; if the equipment proposal is awarded, we will be able to scale this prototype infrastructure by an order of magnitude, dedicate it to computer architecture, and have personnel assigned to managing the Condor pools, developing training/educational material and packaging software.

You can actually try it out yourself. All the software needed to connect to this infrastructure is neatly packaged in a plug-and-play virtual machine (VM) “Grid appliance” which runs on free VMware Player or Server. It usually takes 15-30 minutes to install and start using the software to submit Condor jobs to our pool. This is essentially the procedure that other users will go through once the infrastructure is fully deployed (in the actual infrastructure users will be required to register).

How can I contribute to and use Archer?
The goal of Archer is to be a community resource which will grow in capacity and capabilities as new users join it; in order to contribute to and use Archer all that will be needed is to download and deploy one or more VMs. At this point we are seeking feedback and support for the community to help fund the bootstrap of a seed infrastructure; your contribution at this point would be extremely valuable in helping us show community buy-in. You can give us feedback by:
- Downloading and using an Archer VM Grid appliance
- Following a quick tutorial and providing us with feedback through an anonymous Web survey. Detailed instructions are available at: http://wow.acis.ufl.edu

Needs and usage scenarios:
Modern computer architecture research is driven by quantitative analysis, often requiring the performance of many benchmark applications to be evaluated on several cycle-accurate simulated configurations. Therefore, the lack of access to high-throughput resources that can support the concurrent execution of many long-running jobs is a substantial handicap for researchers in this community. Archer has the potential to enable new achievements in computer architecture research and education, primarily because it will allow faculty and students from institutions that are not equipped with the facilities and staff supporting local high-performance computing to easily tap into its capacity. The following examples illustrate how Archer can fulfill this potential:

Scenario 1: High-throughput cycles for research: Graduate student Maria is preparing a paper on a novel cache design for submission. She has developed a simulator which models her design. However, each simulation takes on average 12 hours to complete on her desktop, and she wishes to analyze 10 configurations on 16 SPEC CPU benchmarks. The time to run this experiment on her desktop is prohibitively large (80 days). She downloads and instantiates an Archer appliance. After developing her code in the VM, she prepares a Condor submit file (building on a tutorial), and queues 160 jobs. The seed Archer resources are utilized at 75% capacity; still, her simulations are expected to finish within a day.

Scenario 2: Local resource pooling and community sharing: A research group has a local set of resources, time-shared and scheduled by students via scripts. Because the scripts do not provide load balancing, often resources become contended. They try out the Archer VM appliance, and decide to join. Interacting with Archer management, they set up a local Condor pool. Their resources are load-balanced, and when not in use, they become available to other Archer users with Condor flocking.

Who is involved in this project?
The proposal is being led by a group of distributed systems and computer architecture researchers. The proposal team includes the University of Florida (Renato Figueiredo, Jose Fortes, P. Oscar Boykin, Tao Li, and Jih-Kwon Peir, with expertise in distributed computing, virtualization, self-configuring and peer-to-peer systems, and computer architecture). Univ. of Wisconsin’s (Alain Roy, from the Condor middleware group), U. Minnesota (David Lilja), UT Austin (Lizy John), Cornell (Sally McKee), Northeastern U. (David Kaeli), FSU (Gary Tyson), Northwestern U. (Gokhan Memik).
What applications will run on this infrastructure?

The system will support batch applications that run on x86/Linux systems. Users will be able to compile and deploy their own research applications, as well as to use typical applications for education (for example, SESC-based or SimpleScalar-based cycle-accurate simulators) that will be made available in application repositories. The scheduler used in the infrastructure will be Condor, which supports checkpointing and migration of applications that are linked with its libraries. Condor also supports the execution of applications that are not linked against its libraries (in which case checkpointing and migration are not supported).

Example: Here is one example of an experiment that was conducted in our prototype Archer. An Archer node running on a desktop spawned a set of 200 Condor SimpleScalar sim-cache jobs to simulate various cache parameters on a SPEC benchmark (go). The median execution time of a single job was 1 hour and 10 minutes, but because there were 56 VMs distributed across five prototype Archer sites (in Florida, Minnesota, Virginia, Louisiana and Illinois), the total time to finish the entire 200-job batch was less than 8 hours. In steady state, jobs were completing at a rate of about one per every three minutes, as shown in Figure 2. The script that created this batch simulation request was only 25 lines in size, and SimpleScalar executed without requiring any source code modifications.

![Cumulative distribution, task completion times](image)

Figure 2: Completion time distribution for a batch of 200 simulation jobs running on a 56-node prototype Archer.

How will the infrastructure be managed?

The setup on each Archer site will entail deploying and keeping Archer VMs operational. The software running on these machines will be free VMware/Xen VM monitors; we will provide installation instructions. Once started, the VMs will form a Condor pool accessible to all other users in the infrastructure. If your site deploys several VMs, they can be organized as a cluster, and jobs submitted from your site will have higher priority in using your local resource, with the capability of preempting remote jobs. Jobs submitted from your site will run in your local pool by default; if the pool is fully utilized, jobs from your site will be able to execute on remote resources using Condor’s flocking technique. Other x86/Linux or x86/Windows machines that you already have at your site can also be added to the proposed local pool of resources where you will have high priority by instantiating the same VM images.

How is this related to efforts such as TeraGrid, PlanetLab, SETI@Home?

The NSF TeraGrid ([www.teragrid.org](http://www.teragrid.org)) is a high-performance infrastructure which provides access to an established set of resources well-suited to run large parallel jobs, as well as high-throughput batch jobs. The proposed infrastructure is different in that it allows individual users to not only access the system, but also contribute with resources. It is also different in that it is focused on needs of the computer architecture community; it will be populated with applications and tutorials that will help architects get started quickly on using the system. Regarding the mechanisms and procedures envisioned to provide access to the infrastructure, in the proposed infrastructure users will be able to sign up and use the system within minutes by booting up a virtual machine that allows them to both submit and execute jobs; in contrast, proposals to obtain allocations of resources on TeraGrid can take several weeks to process.
The proposed system is similar to the Open Science Grid (OSG, www.opensciencegrid.org) in the goal of pooling resources together across multiple institutions; it differentiates from OSG in the mechanisms for adding resources to the infrastructure – in our system, the addition of resources is greatly facilitated by the use of VMs.

The proposed infrastructure is similar to PlanetLab (www.planet-lab.org) in the way that resources can be added to the system by individual researchers who deploy virtual servers in their sites, which are then managed by PlanetLab Central. However, PlanetLab is a generic testbed for experimental networking research and does not support compute-intensive jobs, while our infrastructure targets the execution of compute-intensive applications. Our infrastructure is also different in that it does not require that dedicated physical machines be added to the infrastructure, and in that it provides an isolated virtual private network interconnecting Archer VMs.

Similarly to projects such as ProteinFolding@Home (http://folding.stanford.edu), SETI@home (http://setithome.berkeley.edu), the envisioned system has provisions for new computing nodes to be added seamlessly to the infrastructure. Unlike these projects, where application-specific sandboxes are deployed using screensavers, in our infrastructure the computing nodes are whole-system virtual machines (VMware and/or Xen) with TCP/IP connectivity to an ad-hoc virtual private network, and running the Condor scheduler. Nodes added to the proposed system will be able to both submit and execute simulation applications compiled by its own users.