Involuntary Computing: Hacking the Cloud

Sebastien Goasguen, *Senior Member, IEEE*, Lance Stout and Michael Murphy

***Abstract*—In this short wild and crazy paper we argue that cloud computing has a lot to learn from current cybercriminals who plague the internet with botnets, malware, viruses and extort money from all of us. We present basic distributed systems designs with a view of a hacker and argue that the same designs can be and are indeed used in cloud computing. Decentralized scheduling, network overlay, heterogneous and intermittent resources, redundancy of task execution and asynchronous messaging. All of it via secure channels and using the most widely used protocol http. With this angle we present Kestrel, a job scheduling framework built using XMPP messaging and customized for cloud resources. Our results show that Kestrel scales well over tens of thousands of nodes and we envision a one million agents deployment soon. Finally, we also argue that a botnet is the cheapest cloud with a reported $200 a week for 6,000 cores.**

***Index Terms*—cloud computing, virtualization, botnet, XMPP, wild and crazy idea**

I. Introduction

Volunteer computing was coined in the mid nineties and is best represented by the BOINC middleware [1]. In parallel to volunteer computing, grid computing was developed mostly to support the scientific community in search for increasing computational power. By the end of the nineties another kind of distributed system started to spread: botnets. According to a recent report by Microsoft [2] botnets are currently being rented for an inexpensive cost. The report points out that while early botnets were based on IRC command and control mechanisms, new type of botnets are making use of standard http(s) communication mechanisms (~29%) and a small portion of all botnets (~2.3%) make use of P2P techniques such as Distirbuted Hash Table to build reliable command and control infrastructures [3]. IRC botnets still dominate the market (~38%) but the P2P ones are becoming more common. P2P techniques have been widely covered in theoretical research but its adoption in national grid infrastructure has been extremely slow, while most of the internet traffic is actually caused by P2P based file sharing.

While dealing with cloud resources, which in most cases are virtual machines started on remote sites, we argue that the most difficult challenge is to deal with the networking of these intermittent resources in a world with a shortage of IP adresses. Increasingly cloud resources will be started behind firewall and NATs and will only feature outbound network connectivity. However cybercriminals have long worked in such a hostile environment, in fact they have worked in the toughest of environment. One where the administrator, owner and users of the resources disagree with the intent of the cloud user. Therefore we believe that they have developed very useful technologies and practices to develop middleware that can be of very good use to a non-malicious cloud community.

In particular, we should learn from the cybercriminals, how to deal with adverse network environment, how to scale to a large number of heterogneous workers (millions) which would prove useful with working on supercomputers with 100,000 of cores and how to build a reliable, fault-tolerant system (e.g P2P command and control layer, redundant execution of tasks). Finally, the many infection vectors used (e.g drive by download, USB stick, viruses, worms) present an unorthodox view of cloud “API”, if anything a non-standard one that has proven to be extremely efficient in coalescing various resources. Compared to clouds, botnets are also much cheaper: a reported $200 a week for 6,000 cores. Unfortunately cybercriminals do not benchmark their resources with linpack, nor do they care about latency. In general, bandwidth is a much more important metric for their business (e.g DDoS).

In this paper presentation we will expand on this wild and crazy view that does not adhere to the regular scientific thinking. First we will introduce a new paradigm: Involuntary computing which deals with resource owners that provide cloud resources without being aware of it. Second we will introduce a new cloud specific job scheduling system, Kestrel [4], that leverages cybercriminals concepts using messaging system as a base communication protocol and uses a P2P layer of reliable job managers.

II. Kestrel

Kestrel has been presented in [4] and [5]. It takes its origin in the observation that instant messaging systems have scaled to hundreds of thousands of clients and can provide interactive communications between agents as well as work seamlessly in complex network environment. If a messaging system can scale and work well behind NAT then it can certainly be a solid communication protocol for many nodes in a grid/cloud environment. The second argument for Kestrel is that virtual machines started on cluster may not be allocated public IP adresses. First due to the shortage of IPv4 addresses, second because of the intermittent nature of virtual machines and second for the sheer increase in scale that virtual machines can create via overprovisioning of nodes via consolidation (i.e more virtual machines than cores, can be packed in one physical node).

With these motivating factors, Kestrel has been built on XMPP, formerly known as Jabber, the protocol behind Gtalk and other instant messaging services. We have demonstrated very efficient job dispatching as well as a scale of over ten thousands agents. Current experiments at Clemson University are reaching 40,000 virtual machines running on a production linux cluster.

S. Goasguen is with Clemson University School of Computing, SC, USA (e-mail: <sebgoa,lstout>@clemson.edu, M. Murphy is with Coastal Carolina University, Department of Computer Science, Conway, SC, USA (e-mail: mmurphy2@coastal.edu)

In addition the federation capabilities of XMPP servers provides a strong substrated to create inter-cloud mechanisms. Indeed agents started and reporting to various XMPP servers can either use the same Kestrel manager or a different coupled via database sharing mechanisms. This latest development should enable us to use Kestrel in networks of well over hundred thousand agents and probably millions. Figure 1shows an architecture diagram of Kestrel. Instant Messaging (IM) clients are used to submit and manage jobs, XMPP servers provide the key communication substrate over which Kestrel managers are deployed to dispatch jobs. Workers started on corporate clouds, grid sites or local resources can all join the same Kesrel infrastructure. SSL encrypts the communication between workers and servers. Multiple XMPP servers provide a natural scaling mechanism.

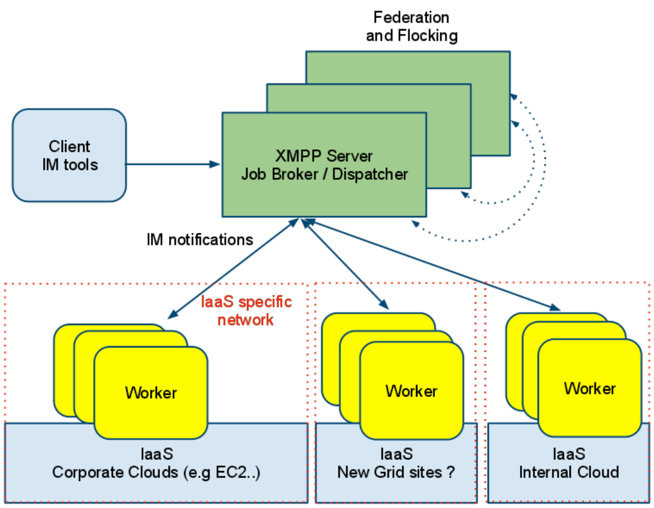

Figure 1. Kestrel [4] Architecture ressembles botnet architecture and leverages the XMPP protocol which has been used by attackers in lieu of IRC channels. Workers started in virtual machines have a Jabber ID and communicate via instant messaging with the managers responsible for job/task scheduling.

Figure 2 shows a snapshot of a standard instant messaging client showing how it can be used to interact with jobs in the Kestrel system. Interaction with jobs is very similar to any standard batch queuing system. A new command line client has also being developed to avoid using a GUI based XMPP client. The STAR collaboration [6] has recently used Kestrel to do one of their largest monte carlo production runs. They used over 400,000 walltime hours on the Clemson Palmetto cluster -a 10,000 cores linux cluster- and ran their jobs for one month straight using Kestrel.

III. Conclusions

While Clouds have seen a terrific uptake in the last few years. Cybercriminals have long borrowed resources over the internet and are now offering on-demand scalable network of resources. At $100 per day for 100,000 bots, the Involuntray clouds appear much cheaper than current corporate cloud providers. In addition, Involuntray cloud APIs are very diverse and instead of converging towards a standard they enjoy a diversity that enables some of the largest cloud. We argue that learning from Cybercriminals we can design better Cloud infrastructure and find new innovative ways to access resources on the internet.

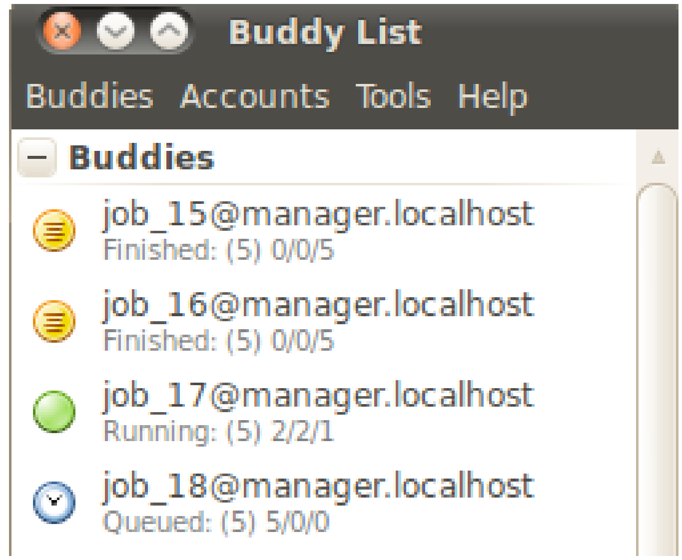

Figure 2. Snapshot of a standard IM client used to control jobs submitted to Kestrel. Each worker has its own Jabber ID (jid). IM commands are used to submit, delete jobs, check the status of the queues and workers. When a worker goes offline Kestrel receives a *worker offline* message.

References

[1] D. Anderson, "BOINC: A system for public-resource computing and storage," in *proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*. IEEE Computer Society, 2004, pp. 4–10.

[2] C. T. Anselmi D., Boscovich R. and C. N., "Microsoft Security Intelligence Report January-June 2010," 2010.

[3] J. Grizzard, V. Sharma, C. Nunnery, B. Kang, and D. Dagon, "Peer-to-peer botnets: Overview and case study," in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*. USENIX Association, 2007, p. 1.

[4] L. Stout, M. Murphy, and S. Goasguen, "Kestrel: an XMPP-based framework for many task computing applications," in *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers*. ACM, 2009, pp. 1–6.

[5] L. Stout, M. Fenn, M. Murphy, and S. Goasguen, "Scaling virtual organization clusters over a wide area network using the Kestrel workload management system," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. ACM, 2010, pp. 692–698.

[6] J. Harris *et al.*, "The STAR experiment at the relativistic heavy ion collider," *Nuclear Physics-Section A*, vol. 566, pp. 277–286, 1994.