Extending Hypervisor Architecture to Allow One Way Data Transfers from VMs to Hypervisors

Mustafa Aydin and Jeremy Jacob
Department of Computer Science, University of York, York, U.K.

Keywords: Cloud, Hypervisor, Forensics, Security, Energy.

Abstract: We propose an alternative architecture to existing hypervisors, which allows for more data to be moved whilst requiring less work for hardware and networks. Our suggestion is to develop an extension to hypervisors for an interface which can allow data transfer one way from virtual machines to hypervisors. We argue that the ability to transfer data directly in this way can provide a number of benefits to cloud users and providers, namely in the areas of security (confidentiality, integrity, and through decreased overhead), reduced energy consumption, and better use of hardware resources.

1 INTRODUCTION

Cloud computing has started to enjoy widespread use, with the convenience of on-demand IT resources proving to be increasingly popular among its users. Although cloud customers are able to employ their machines in much the same way as they can local physical machines, the fact that cloud services are delivered remotely has forced many users to rethink the way in which certain tasks are performed. Among these tasks for example, is the need to back up important data.

The backing up of data is a common operation in any system, and in cloud systems backing up has generally meant redirecting data to secure locations elsewhere. Data backup and redirection may be done for a variety of reasons. Some of the more common reasons include the backing up of log data for security purposes, and the need to backup valuable information which is too precious to lose. The fact that cloud providers immediately delete VM images upon VM termination is one example of why cloud users may wish to redirect data before they no longer have access to it. For this reason redirection of important data is used as an important component of research looking into providing ways to aid cloud forensics investigations (Marty, 2011) (Zawoad et al., 2013).

The fact that cloud computing is becoming more popular means that the need to redirect data will also need to increase. Along with an increase in the amount of data being produced by applications generally, this means the amount of information needing to be backed up and/or redirected is getting larger all of the time. Sending this data will have an effect on performance and network overhead, and may even result in associated costs from the cloud provider, depending on the service used. In a world in which more and more data and metadata is being produced, we need a way in which data can be sent around or redirected in more efficient ways than we currently have at the moment.

Our aim in this work is to provide more details of our suggested design for an extended hypervisor architecture, outlining the advantages of using our approach and responding to possible concerns. Our motivation is provided by the belief that the implementation of our approach would help to make better use of available resources, and also make it more feasible for larger amounts of information to be redirected for use whenever required.

Our paper is structured in the following way: we describe the area of virtualisation in Section 2, we cover the design principles of our suggested architecture in Section 3, and the advantages provided by our approach in Section 4. We look at possible problems in Section 5, discuss other related work in Section 6, and provide a conclusion in Section 7.

2 VIRTUALISATION

Virtualisation is one of the key enablers of cloud computing, allowing a way of operating multiple in-

Aydin M. and Jacob J..
Extending Hypervisor Architecture to Allow One Way Data Transfers from VMs to Hypervisors.
DOI: 10.5220/0004958006040609
In *Proceedings of the 4th International Conference on Cloud Computing and Services Science* (CLOSER-2014), pages 604-609
ISBN: 978-989-758-019-2

stances of operating systems (known as virtual machines, or VMs) on a single physical machine. VMs are run by a more privileged software layer known as a hypervisor, which controls and allocates access to the hardware resources between VMs. Virtualisation has many benefits, providing secure isolation between VMs, and allowing much more efficient use of physical machines by utilising spare hardware capacity. The need to make hypervisors use hardware ever more efficiently has meant the development of techniques which allow them to use available resources better.

One of the biggest developments in this area was the introduction of paravirtualisation. This is a technique in which the hypervisor presents guest machines with an interface through which certain executable commands are directly passed to the hypervisor, commands able to be performed much more quickly than they would if passed through the layers included within the virtual machine. In order to do this, both the hypervisor and the guest operating system must be configured to take advantage of this ability to paravirtualise. By cutting out unnecessary movement of calls through different layers of the virtualised environment, paravirtualisation allows for better utilisation of the hardware.

The successive adoption of paravirtualisation suggests that using effective cooperation between hypervisors and operating systems in order to speed up useful operations can see widespread use. Although paravirtualisation has so far been concerned with allowing operating system executions to be accelerated, there is no reason why the principle of providing faster services cannot be adapted to be useful in other contexts. If the production and subsequent transfer of data is an operation using more resources than necessary, it makes sense to be able to reduce the impact of this where possible.

We believe that taking advantage of similar principles to paravirtualisation may be one way to create a more efficient environment for the cloud. The ability to pass data directly from a VM to a hypervisor could reduce the amount of overall work required compared to sending the same data to a destination over a network. This can be seen in Figure 1, which shows data going through multiple virtual and hardware layers, before reaching its final destination. The forwarded data is first sent to the VM from the application, before being sent off to the outside world for storage. In contrast, in our proposed architecture, the data is sent directly to the hypervisor and VM in parallel, and reaches its destination having gone through as few layers as possible. This would help cloud users and cloud providers to make better use of the available resources, and reduce both performance and network overhead in particular instances.

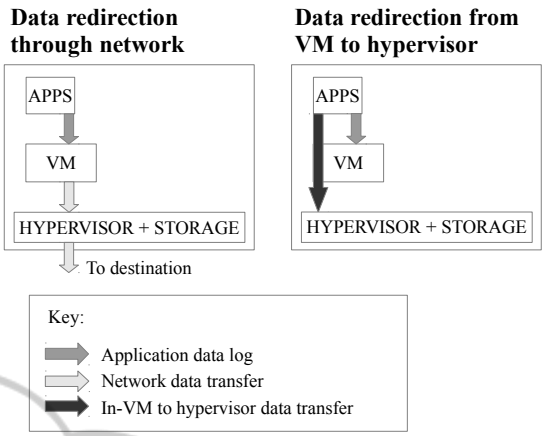

Figure 1: The diagram on the left shows the current architecture, while that on the right shows our proposal.

3 DESIGN

In this section we consider some of the important design principles, in order to provide a better understanding of the way in which the suggested architecture would work.

3.1 Storage

Redirecting data over the internet requires the data to go through many virtual and physical layers. In our proposed hypervisor extension, the data can be redirected directly to the hypervisor layer on the same physical machine (see Figure 1). Instead of passing through all of those layers and causing the related performance and network overheads, the data can be redirected with minimal effort to an environment which is already trusted by the user.

The location of storage of the transferred data is intended to be on the hypervisor layer, in a partition which is linked to the running VM (and associated user). This will allow subsequent retention of the information even when the VM has been terminated, and can be moved to a more convenient place after the VM has been shutdown. The ability to access the data will need to be offered to the user, but this may be of more concern to the cloud provider than to the design of the extended hypervisor.

3.2 Privacy

Privacy is an important concept to cloud users, who want assurances that their running VMs and any as-

sociated data is accessible only to themselves. Users want to have the option of privacy for all data they intend to send to alternative locations in the cloud. The use of techniques such as encryption is one way to provide this assurance.

Encryption of data at the moment of creation (*i.e.* before being passed to the hypervisor) would be a useful way of doing this. Users would have a level of assurance that even an administrator would be unable to access the transferred data (unless they somehow learned the encryption key). Techniques which allow the hypervisor to monitor some of the high level activity within a VM do actually exist, such as virtual machine introspection (Garfinkel and Rosenblum, 2003). However without the ability to monitor low level OS calls to the processor, the only other way someone would be able to break the encryption would be poor security on the part of the user.

Using established encryption techniques within the interface for data transfer should be enough to satisfy users that they have strong privacy in place for their data.

3.3 One Way

The interface for data transfer is intended to be only one way, allowing for the transfer of data from the virtual machine to the hypervisor. The reason for this limitation is that there is no conceivable reason for the hypervisor to send any data to the virtual machine, other than passing the communications which are required to be managed between the OS and the hardware.

Preventing the ability to send information from the hypervisor to the VM would be a necessary safety measure. Communication from hypervisors to VMs would provide significant security worries for users, and make hypervisors targets for hackers. Attackers would try to use any such abilities to attack VMs from the hypervisor level. Threats could even try to make use of such opportunities from within VMs, by communicating with other VMs on the same hardware to coordinate attacks. Maintaining the isolation of VMs is an important security principle which could be violated by the ability to have two way messaging.

A problem created by the need to have one way messaging would be the lack of acknowledgement from the hypervisor for transferred data, which may result in some incorrect data transfers.

3.4 Integrity

Integrity frequently comes into question after data moves location. Redirecting data across the cloud is an obvious case of when data integrity needs to be verified. It may be less obvious that it would need to be done when moving data from a VM to a hypervisor, since it does not move off the physical machine. However the data could always move later on, in which case its integrity would still need to be verified. In addition, in the cloud the hypervisor layer does not belong to the cloud user, making integrity an important feature of the data.

The interface used for sending data to the hypervisor needs to also have the ability to sign information with keys, in order to satisfy this need. Quicker signing methods using single keys would be useful to keep the performance overhead down, but more complicated ones using changing keys could also be considered depending on a user's preferences. Doing this at the hypervisor level could be problematic, since there may need to be a requirement to manage different keys for VMs, since a single physical machine in a cloud environment is likely to be providing services to more than one customer. It would make more sense for each operating system to be able to manage its own keys, reflecting the need for both hypervisors and operating systems to work well together in order to make better use of their environment.

3.5 Trust

The design presented here is based on the assumption that users trust the hardware owner of the hypervisor they are using. There is no more trust for the presented hypervisor architecture than is needed in any other proprietary hypervisor. The point here is only to reinforce the idea that whenever hypervisors are being used, there is a level of trust placed in the hardware owner the moment that a user chooses to use a VM on that platform.

Future possibilities for those who do not trust the hardware owner may lie in the use of homomorphic encryption technology, in which fully encrypted data is still able to be processed without the need for decryption (Gentry and Halevi, 2011). Use of such methods could be useful in order to keep valuable information safe on untrusted environments like the cloud, although the problem of its slow running speed will need to be solved.

4 ADVANTAGES

In this section we consider possible advantages from using the proposed architecture.

4.1 Speed

One of the most obvious benefits of the proposed architecture is the fact that the delivery of data to the intended destination would be performed as quickly as the hardware would allow. Delivery of data to any other location would be dependant on networks and network layers, and hardware and hardware layers. Even though data may always succeed in getting to the intended destination, services which would require fast delivery would still have to wait for data to be sent. For security monitoring processes, the difference in the time taken to do this could be important.

4.2 Overhead

Overhead can be broken down into two main areas, performance overhead and network overhead. The redirection of data to alternative locations affects both of these. Even copying a file onto another location on a local hard disk can noticeably slow down a running machine, so any opportunity with which this kind of costly data transfer can be avoided would be preferable to the average user.

An interface able to move data from one layer to another could also be used to specify a duplicate location in the hypervisor layer to place data at the moment it is created in the hardware. The simultaneous writing of the data to both locations avoids the need to copy any data later on, so the job does not need to be performed a second time (unless required for another reason).

The fact that data would no longer necessarily need to be sent over a network would relieve the pressure on networks too. This would allow for virtual machines which have a need to preserve network capacity to use their network speed for purposes other than the back up of data.

4.3 Security

Although sending information over networks is not inherently insecure if done correctly, keeping data within a trusted boundary does have its advantages. The ability to send data directly to another level on the same hardware prevents attacks such as a man in the middle attack from taking place, a worry for those with particularly sensitive information.

The most obvious point of attack for the VM owner's data is by a malicious administrator, who would be able to access the VM owner's data and their VM image, but would not be able to access them without also gaining the passwords with which they were encrypted. However, this danger would exist anywhere on the cloud on which a user stores data. By keeping data within trusted confines, it would at least reduce the need to look for additional services which can offer third party storage.

4.4 Energy

Energy use has become a much more important concern for organisations, as they have started to become aware of the costs of its use. Energy costs will be of most concern to cloud providers, due to the huge amounts of energy required to power their data centres. There are ways in which this concern could be passed to users as well. Firstly, if cloud providers ever decide to start using energy units as part of the cost calculations for their services. Secondly, due to wider concern about energy use, it is not unlikely that cloud providers could start to provide a breakdown of the energy use of individual virtual machines.

Work looking at the energy use of different kind of cloud services, and measuring the energy used to send cloud data around the world, is now being produced (Schien et al., 2012). Our suggested architecture could provide a way to decrease overall energy use in the cloud, by reducing the need to send data over the various hardware and network layers (which for users who redirect huge amounts of data could amount to a significant saving). Even for users who redirect small amounts of data, this could add up to a worthwhile reduction in energy use (especially over time, or from a collection of machines). With energy consumption becoming an environmental concern too, this is the kind of approach regular cloud users may express an interest in too.

5 PROBLEMS

Although our discussed design is intended to reduce issues, there is still the possibility that there will be problems.

5.1 Malware

In theory, malware might be able to use the opportunity to pass data directly to a hypervisor as an ideal opportunity to launch an attack. Despite this worry, hypervisors have so far proven to be well isolated against attack from VMs.

Nevertheless with security threats constantly evolving, there may develop new ways in which cleverly designed malware could use the access to the hypervisor provided here to attack the host, and by extension, the running VMs on the host. As an addi-

tional precaution therefore, it would be prudent to add additional security by writing the transferred data to a space from which nothing should be able to execute.

5.2 Scalability

Although one of the main benefits of the intended system design is to reduce the overall amount of work performed for the delivery of data overall, it is unknown exactly how the system would perform with large amounts of data being sent from multiple instances. These kinds of problems would only be able to be analysed from a prototype, from which any issues could then be properly addressed to make the architecture better able to deal with them.

5.3 Vendor Lock-in

Vendor lock-in is the problem of use of a technology or service by a customer from which they cannot easily transition to a competitor's alternative offering. There are already problems around this issue for various cloud services, such as the inability to move VMs between different cloud providers.

In the architecture we have outlined here, it is clear that users who would want to send data to the hypervisor layer, and hope to store that data within their user account, would have to do it with their VM provider. The only alternative ways of redirecting data would have to involve redirecting over the network.

6 RELATED WORK

Apart from the development of paravirtualisation, work on extended or modified hypervisors making use of alternative architectures have received some attention in the research literature.

Garfinkel *et al*. developed Terra, an architecture designed to help increase VM and application security (Garfinkel et al., 2003). Terra provides users with the ability to run applications within regular "open-box" VMs or specially protected "closed-box" VMs (with tailored security requirements), and uses Trusted Computing to authenticate communications betweeen applications.

Sailer *et al*. developed a secure hypervisor architecture called SHype, specifically to control data flows between VMs (Sailer et al., 2005). This was achieved by implementing Mandatory Access Control to enforce security policies from the hypervisor level in addition to relying only on OS security controls like a regular hypervisor would.

Keller *et al*. propose the NoHype architecture in order to reduce the risk of attack to the hypervisor layer (Keller et al., 2010). Instead of a hypervisor the system uses hardware features to split access to each of the required resources (in a similar way to a VMM) for each guest OS by utilising extensions to existing hardware.

7 CONCLUSION

We have suggested an interface to allow data transfer to take place from VMs to hypervisors, as a way of reducing the overall amount of work associated with data transfer over the cloud. We have looked at the problems associated with transferring data in this way, and demonstrated why it may be preferable to be able to send them directly to the hypervisor layer. By reducing the amount of overhead needed to perform particular jobs, and allowing users to transfer more data in an easier manner, we believe that such an architecture would be able to offer improved security, better utilisation of hardware resources in the cloud, and help to reduce energy consumption.

In order to do this, both operating systems and hypervisors would need to be adapted to take advantage of this property. The success of paravirtualisation suggests that cooperation between OS designers and hypervisors designers can make such interfaces work well.

The next step for this work will be to develop the extended architecture and to test with a suitable operating system. Validation through experimental results could encourage cloud stakeholders that the presented design is worth using to provide cloud services.

ACKNOWLEDGEMENTS

The authors thank BT for their contribution to this work, and the reviewers for their suggested corrections. Mustafa Aydin is sponsored by BT and is an EPSRC funded EngD student, and part of the LSCITS project.

REFERENCES

Garfinkel, T., Pfaff, B., Chow, J., Rosenblum, M., and Boneh, D. (2003). Terra: A virtual machine-based platform for trusted computing. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP '03, pages 193–206, New York, NY, USA. ACM.

Garfinkel, T. and Rosenblum, M. (2003). A virtual machine introspection based architecture for intrusion detection. In *In Proc. Network and Distributed Systems Security Symposium*, pages 191–206.

Gentry, C. and Halevi, S. (2011). Implementing gentrys fully-homomorphic encryption scheme. In *Advances in Cryptology–EUROCRYPT 2011*, pages 129–148. Springer.

Keller, E., Szefer, J., Rexford, J., and Lee, R. B. (2010). Nohype: Virtualized cloud infrastructure without the virtualization. *SIGARCH Comput. Archit. News*, 38(3):350–361.

Marty, R. (2011). Cloud application logging for forensics. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 178–184, New York, NY, USA. ACM.

Sailer, R., Valdez, E., Jaeger, T., Perez, R., Doorn, L. V., Griffin, J. L., Berger, S., Sailer, R., Valdez, E., Jaeger, T., Perez, R., Doorn, L., Linwood, J., and Berger, G. S. (2005). shype: Secure hypervisor approach to trusted virtualized systems. In *IBM Research Report RC23511*.

Schien, D., Preist, C., Yearworth, M., and Shabajee, P. (2012). Impact of geographic location on the energy footprint of ict services. In *IEEE International Symposium on Sustainable Systems and Technology (IEEE ISSST 2012)*. Conference Organiser: IEEE.

Zawoad, S., Dutta, A. K., and Hasan, R. (2013). Seclaas: Secure logging-as-a-service for cloud forensics. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 219–230, New York, NY, USA. ACM.