SURVEY ON FINDING FAULTS USING VIRTULIZED DATACENTERENVIRONMENTAND VIRTULIZATION FAULT TOLERANCE APPROACH TOWORDS MANDATORY SECURITY MONITORING

Kshamarani Purvimath[1], Shriharsha S.Veni[2]

[1]PG Scholar, [2]Assistant Professor, Department of Computer Science & Engineering ,BLDEA's Dr.P.G.Halakatti College of Engineering and Technology, Vijayapura, Karnataka, (India)

ABSTRACT

Virtualized datacenter (VDC) has become a popular approach to large-scale system consolidation and the enabling technology for infrastructure-as-a-service cloud computing. Fault tolerance in cloud computing is a grand challenge problem now a days. The main fault tolerance issues in cloud computing are detection and recovery. To combat with these problems, many fault tolerance techniques have been designed to reduce the faults. Virtualization and Fault Tolerance (VFT) technique is used to reduce the service time and to increase the system availability. The VFT technique and the VDC technique are widely used in cloud computing to keep the user's data secure. Malware is one of the most serious security threats on the Internet today. In fact, most Internet problems such as spam e-mails and denial of service attacks have malware as their underlying cause. The security monitoring would either be up to the discretion of individual tenants or require costly direct management of guest systems by the VDC operator. We propose the EagleEye approach for on-demand mandatory security monitoring in VDC environment which does not depend on pre-installed guest components. This survey paper focuses on both fault tolerance approach and EagleEye approach in cloud computing platforms and more precisely on autonomic repair in case of faults. In most of current approaches, fault tolerance is exclusively handled by the provider or the customer, which leads to partial or inefficient solutions. Solutions, which involve collaboration between the provider and the customer, are much promising. We propose the EagleEye approach for on-demand mandatory security monitoring in VDC environment, which does not depend on pre-installed guest components. We implement a prototype on access anti-virus monitor to demonstrate the feasibility of the EagleEye approach. We also identify challenges particular to this approach, and provide a set of solutions meant to strengthen future research in this area.

Keywords: VM,VDC, VFT, EE, CSP, CM, DM

I. INTRODUCTISON

Virtualization is an emerging IT paradigm that separates computing functions and technology implementations from physical hardware. Virtualization technology allows servers and storage devices to be shared and utilization be increased. Applications can be easily migrated from one physical server to another. Virtualized datacenter is a pool of cloud infrastructure resources designed specifically for enterprise business needs. Those resources include compute, memory, storage and bandwidth. A virtual datacenter in the context of cloud computing services falls within the infrastructure-as-a-service (IaaS) category. It enables you to quickly access cloud infrastructure from a service provider such as bluelock. The security monitoring on critical system in the environment as a means to track and deter the threats that could jeopardize the operation of the VDC.

Fault tolerance is an approach where a system continues to success even if there is a fault. Although there are number of fault tolerant models or techniques are available but still fault tolerance in cloud computing is a challenging task. Because of the very large infrastructure of cloud and the increasing demand of services an effective fault tolerant technique for cloud computing is required. In this survey paper fault tolerance is integrated with the cloud virtualization As shown in the figure 1. Our fault tolerance is a kind of reactive fault tolerance approach.

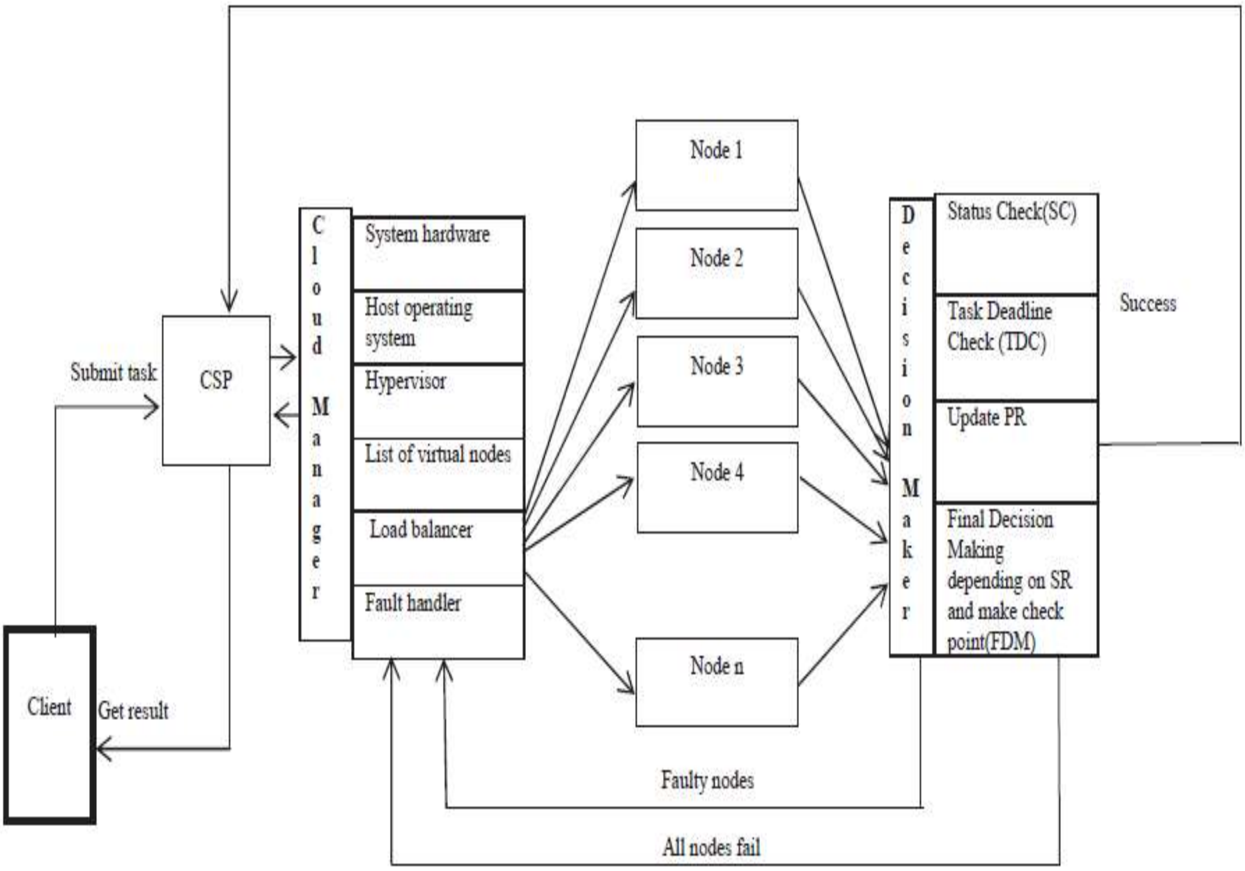

Fig 1: Proposed VFT Model

The basic mechanism to achieve the fault tolerance is replication or redundancy. We have performed this replication in form of software variants running on multiple virtual machines.We have presented a virtualization approach with the help of hypervisor where the load balancer takes high responsibility by distributing loads only to those virtual nodes whose corresponding physical servers have a good performance history. To measure the performance history of a physical server we have used success rate. If n1=number of times a physical server gives successful results and n2=total number of times requests sent to that server, then the Success Rate SR =n1/n2, where n1<=n2. Malicious code, or malware, is one of the most pressing security problems on the Internet. Today, millions of compromised web sites launch drive-by download exploits against vulnerable hosts. As part of the exploit, the victim machine is typically used to download and execute malware programs. These programs are often bots that join forces and turn into a botnet. Botnets are then used by miscreants to launch denial of service attacks, send spam mails, or host scam pages. Here we have the Virtualization and Fault

Tolerance (VFT) model and this proposed model provides the reactive fault tolerance on cloud infrastructure. This scheme tolerates the faults on the basis of Success Rate (SR) (0<SR<=1) of each virtual node's physical server. A virtual node is selected for computation on the basis of SR of its corresponding physical server and can be removed, if the selected node's physical server does not perform well. Our model consists of two main modules Cloud Manager (CM) module and Decision Maker (DM) module.

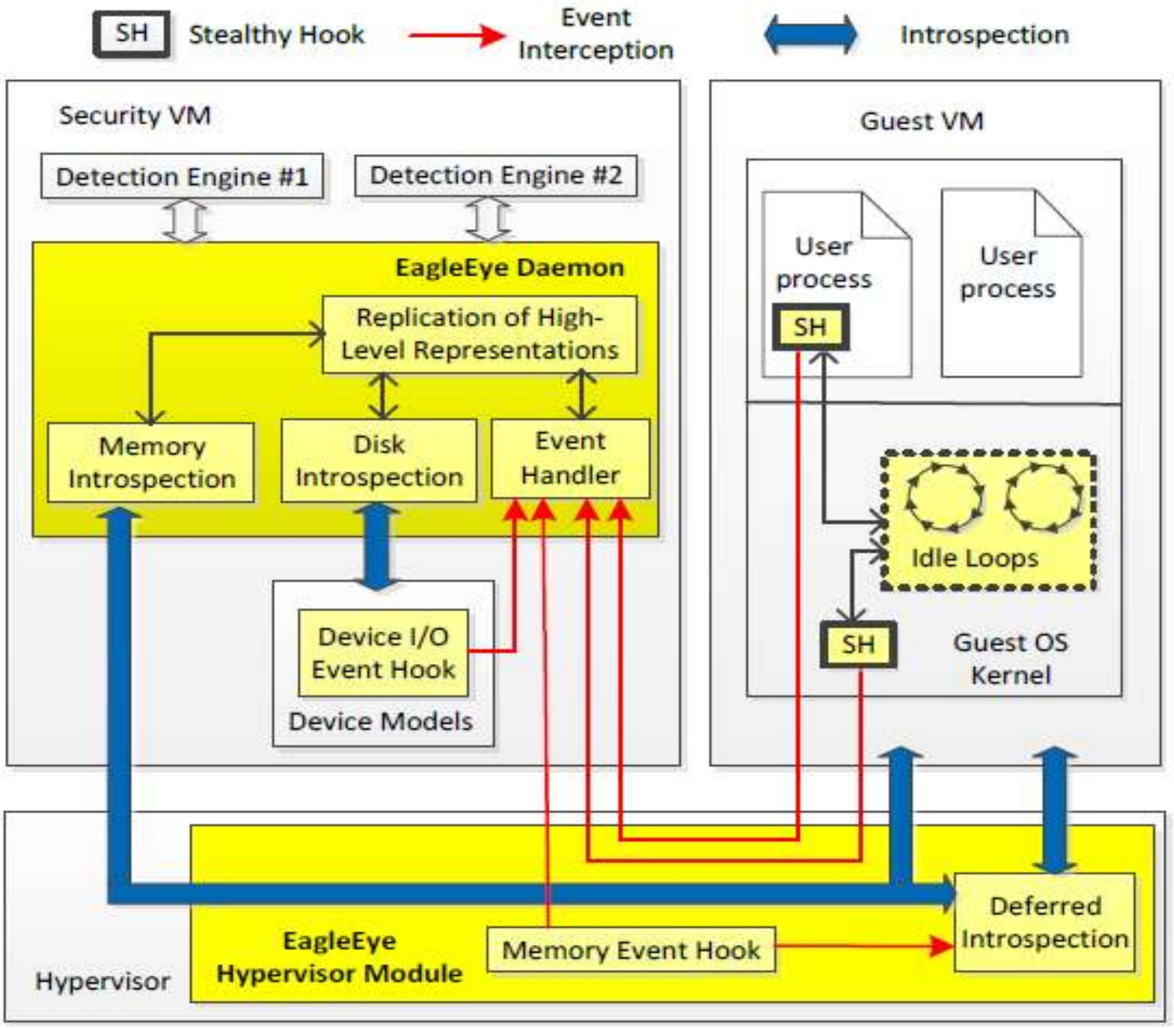

Figure 2: Eagle Eye Architecture

Cloud Manager (CM) is included in the cloud architecture. It performs the virtualization with the help of Hypervisor. Hypervisor is a low level program which creates a virtual environment and provides system resource access to virtual machines. When virtual nodes are created from the available resources of the physical servers (System Hardware) then Hypervisor maintains a record of which virtual node belongs to which physical node. Resources of a single physical server can be used to create set of virtual nodes. A Performance Record (PR) table is maintained containing the server's ids, virtual nodes ids and Success Rate (SR) to identify the virtual nodes and to keep record of the number of times tasks are assigned to the virtual nodes number of customized VMs in a VDC it will be a quite expensive process for a VDC operator to deploy and manage security monitors in each of the VMs. In addition, VMs in a large-scale VDC are often managed by individual tenants and not by the datacenter operator. One will have to rely on individual tenants to deploy and manage the security monitors in their respective VMs. Obviously, this approach is problematic since a negligent tenant can inadvertently disable the security monitor, and a malicious tenant may even attempt to tamper with the security monitor. Motivated by the above difficulties, we propose the EagleEye mandatory security monitoring approach for VDC environment as shown in the figure 2. In the approach, security monitors are placed externally to the guest VMs. There is no requirement for installing guest components in the VMs. It requires no attention or cooperation from the VM tenants. The approach also allows automated deployment and management of security monitors in a VDC environment. To demonstrate the feasibility of the proposed approach, we built a prototype on-access malware detection system for guest VMs in a VDC. Achieving transparent guest system event interception, resolving inconsistent guest states during synchronous security monitoring, bridging the semantic gap across complex blackbox guest system models, and reducing the performance overhead of blocking-wait in the synchronous monitoring mode. In case of Virtualization and Fault Tolerance technique we have used the Success Rate computation algorithm and Decision technique algorithm to achieve the good performance of nodes.

1.1 Success Rate Computation Algorithm

```
1. Initially success rate =0.5, n1=1, n2=2
2. n1 is the number of times the virtual node of a particular physical server gives successful results
3. n2 is the number of times the Load Balancer of the cloud manager(CM) assigns tasks to a particular server’s virtual node
4. Input maxSuccessRate=1
5. Status of a node is Success if SC and TDC module for that node is success
6. Status of a node is Fail if SC or TDC or both module for that node is fail
7. if (nodeStatus = =Success) /*SC and TDC success */
{
n1=n1+1
n2=n2+1
SuccessRate = n1/n2
Update PR table
}
Else
{
if( nodeStatus = =Fail ) /* SC or TDC or both fail */
{
n2=n2+1
SuccessRate= n1/n2
Update PR table
}
}
8. if (SuccessRate >= maxSuccessRate)
{
SuccessRate = maxSuccessRate
}
9. if(SuccessRate<=0)
```

```
{
Reject the node and inform the Cloud
Manager to add a new node
}
```

1.2 Decision Technique Algorithm

```
1. Initially SuccessRate=0.5
2. Input from TDC: node_SuccessRate, n=no.of nodes with SC and TDC success
3. Input maxSuccessRate
4. if(n= =0)
{
Status = fail
Perform backward recovery with the last successful checkpoint
}
5. else
{
Status =Success
Best Success Rate = find Success Rate of node with highest SuccessRate
Select the node with bestSuccessRate and send the result to CSP
Make checkpoint
}
```

II. LITERATURE SURVEY

A lot of work has been done in the area of fault tolerance for cloud computing. But due to its virtualization and internet based service providing behaviour fault tolerance in cloud computing is still a big challenge. Many researchers have given various fault tolerance techniques and strategies in [4], [5], [6], [7], [8], [9], [10], [11], [12], [20] , [21], [24] and in [25]. Dilbag Singh and Jaswinder Singh in [4] have given failover strategies for cloud computing using integrated checkpointing algorithms. Sheheryar Malik and Fabrice Huet in [10] have given an approach for adaptive fault tolerance in real time cloud computing. Our proposed model not only tolerate faults but also reduce the chance of future faults by not assigning tasks to virtual nodes of physical servers whose success rates are very low. The concept of VMM based security monitoring was proposed by Garfinkel et al [6]. Their security monitor can perform integrity check of the guest kernel and programs and can also detect NIC promiscuous mode usage. The semantic gap problem in VM introspection was discussed in XenAccess [10, 18], VMwatcher [26], and Virtuoso [19]. However, none of the above work can be used to deal with semantic gaps caused by complex mechanisms such as disk caching. Event-driven VMM monitoring was proposed in the system Lares [26]. Lares employs a PV driver in a guest VM to reroute events of interest to an external security application. VMware provides a set of introspection API called VMware [24] for security monitoring on VMware platform. The API allows the introspection of guest VM network, CPU, memory, and disk storage states. Event-driven monitoring is supported through a PV driver (i.e. the vShield endpoint driver). VMsafe has been employed in products such as Trend Micro Deep Security and McAfee MOVE. Our work is distinct from these works in that our approach does not require PV drivers to hook and reroute the guest events.

Virtualization-based monitoring has also been applied to dynamic malware analysis [15, 25]. The motivation is that hardware assisted virtualization can be leveraged to hide the analyzer. The analysis environment is purposely built and not part of a production system, so issues such as overall system performance and deployment cost are not as relevant as in the realm of online security monitoring. Also, malware analysis system focuses more on extracting the full behavior of a malware. The analysis does not have to be synchronous and responsive. It can assume that complete information about the system and the malware under analysis can be acquired at a later time. On the contrary, security monitoring often has to make monitoring decisions synchronously and immediately based on very limited information at that time point. Rosenblum et al. [26] first proposed the use of virtualization to separate instruction execution and data access contexts on memory pages. We adopt the same strategy of memory context separation to hide the stealthy hook from guest detection. However, our implementation takes advantage of the extended page table virtualization hardware and does not require every guest page fault to be trapped into the hypervisor.

III. CONCLUSION

Although a considerable amount of research effort has gone into malware analysis and detection, malicious code still remains an important threat on the Internet today. Unfortunately, the existing malware detection techniques have serious shortcomings as they are based on ineffective detection models. This survey paper proposes a smart failover strategy for cloud computing using success rate of the computing nodes and virtualization which include the support of load balancing algorithms and fault handler. Performance comparison of existing methods has been made with the proposed method. It has been concluded with the help of performance metric’s comparison and success rate analysis from simulated results that the proposed fault tolerant strategy gives a very good performance. In our future work we will work on the Fault Handler and Load Balancer sub modules of CM module in order to make the model more fault tolerant. We propose the EagleEye approach to achieve mandatory security monitoring in virtualized datacenter environment. The approach has been applied to a real-world security monitoring application. In EagleEye, we come up with the technique of high-level representation replication to address the semantic gap and the inconsistent system state problems. The technique is powerful enough to deal with complex black-box mechanisms such as disk caching. The requirement for synchronous monitoring is supported by the stealthy hook mechanism, which is transparent (to the guest) and scalable. We proposed the deferred introspection technique as an enhancement of memory introspection to deal with inconsistent guest memory states due to on-demand paging or memory swapping. The goals of mandatory security monitoring prevent the use of guest kernel synchronization mechanisms to implement efficient blocking wait for security monitoring. The strength of EagleEye being able to operate without a PV-driver is also its weakness.. We look forward to the community engaging in dialog that would help mature the technologies.

REFERENCES

[1] Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013), [1]Pranesh Das, [2]Dr.Pabitra Mohan Khilar ,Department of Computer Science and Engineering,National Institute of Technology ,Rourkela-769008.

[2] Yu-Sung Wu*, Pei-Keng Sun, Chun-Chi Huang, Sung-Jer Lu, Syu-Fang Lai and Yi-Yung Chen,Department of Computer Science ,National Chiao Tung University, Taiwan

hankwu@g2.nctu.edu.tw, bacon.cs99g@nctu.edu.tw, tracyttin.cs00g@nctu.edu.tw, blackxwhite@gmail.com, cheniy.cs97@nctu.edu.tw

[3] Clemens Kolbitsch_, Paolo Milani Comparetti_, Christopher Kruegel‡, Engin Kirda§, Xiaoyong Zhou†, and XiaoFeng Wang†, § Institute Eurecom, Sophia Antipolis kirda@eurecom.fr † Indiana University at Bloomington ,{zhou,xw7}@indiana.edu

[4] Dilbag Singh, Jaswinder Singh, Amit Chhabra, “High Availability of Clouds: Failover Strategies for Cloud Computing using Integrated Checkpointing Algorithms”, IEEE International Conference on Communication Systems and Network Technologies, 2012.

[5] G. A. Gibson, B. Schroeder, and J. Digney, “Failure Tolerance in Petascale Computers”, CTWatchQuarterly, vol 3,no 4, November 2007.

[6] Ifeanyi P. Egwutuoha, Shiping Chen, David Levy, Bran Selic, “A Fault Tolerance Framework for High Performance Computing in Cloud”, 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012.

[7] Ghassem Miremadi, Johan Karlsson, Ulf Gunnejlo, Jan Torin,” Two Software Techniques for On-line Error Detection”, Twenty-Second International Symposium on Computing & Processing (Hardware/Software), 10.1109/FTCS.1992.243622.

[8] Dimitris Gizopoulos, Mihalis Psarakis, Sarita V. Adve, Pradeep Ramachandran, Siva Kumar Sastry Hari, Daniel Sorin, Albert Meixner, Arijit Biswas, and Xavier Vera, “Architectures for Online Error Detection and Recovery in Multicore Processors”, Proceedings of the Design, Automation and Test in Europe (DATE), March 2011.

[9] Anju Bala, Inderveer Chana ,”Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing”, IJCSI International Journal of Computer Science Issues, vol. 9, Issue 1, no 1, January 2012.

[10] Sheheryar Malik, Fabrice Huet, “Adaptive Fault Tolerance in Real Time Cloud Computing”, IEEE World Congress on Services, 2011.

[11] Wenbing Zhao, P. M. Melliar-Smith and L. E. Moser, “Fault Tolerance Middleware for Cloud Computing”, IEEE 3rd International Conference on Cloud Computing, 2010.

[12] Ekpe Okorafor, “A Fault-tolerant High Performance Cloud Strategy for Scientific Computing”, IEEE International Parallel & Distributed Processing Symposium, 2011.

[13] B. Payne. (2012, 1/12). LibVMI. Available: http://vmitools.sandia.gov/libvmi.html

[14] A. Dinaburg, P. Royal, M. Sharif, and W. Lee, "Ether: malware analysis via hardware virtualization extensions," in ACM CCS, 2008, pp. 51-62.

[15] B. D. Payne, M. de Carbone, and W. Lee, "Secure and flexible monitoring of virtual machines," in ACSAC, 2007, pp. 385-397.

[16] B. Dolan-Gavitt, T. Leek, M. Zhivich, J. Giffin, and W. Lee, "Virtuoso: Narrowing the semantic gap in virtual machine introspection," in IEEE Sympoisum on Security and Privacy, 2011, pp. 297-312.

[17] X. Jiang, X. Wang, and D. Xu, "Stealthy malware detection through vmm-based out-of-the-box semantic view reconstruction," in ACM CCS, 2007, pp. 128-138.

[18] A. M. Nguyen, N. Schear, H. D. Jung, A. Godiyal, S. T. King, and H. D. Nguyen, "Mavmm: Lightweight and purpose built vmm for malware analysis," in ACSAC, 2009, pp. 441-450.

[19] N. E. Rosenblum, G. Cooksey, and B. P. Miller, "Virtual machine provided context sensitive page mappings," in International Conference on Virtual Execution Environments, 2008, pp. 81-90.

[20] Andreas Menychtas, Kleopatra G. Konstanteli, “Fault Detection and Recovery Mechanisms and Techniques for Service Oriented Infrastructures”, 2012, IGI Global.

[21] Yi Hu, Bin Gong, Fengyu Wang, “Cloud Model-Based Security-aware and Fault-Tolerant Job Scheduling for Computing Grid”, The Fifth Annual ChinaGrid Conference, IEEE, 2010.

[22] N. Yadav and P.M.Khilar, “Hierarchically Adaptive Distributed Fault Diagnosis in Mobile Adhoc Networks using Clustering”, in proc. Of International Conference on Industrial and Information System – 2010 (ICIIS 2010), NITK, Surathkal, Karnatak, India during 29th July to 1stAugust 2010.

[23] Pabitra Mohan Khilar, Jitendra Kumar Singh, Sudipta Mahapatra,“Design and Evaluation of a Failure Detection Algorithm for Large Scale Ad Hoc Networks Using Cluster Based Approach”, IEEE International Conference on Information Technology,10.1109/ICIT.2008.72.

[24] B. D. Payne, M. Carbone, M. Sharif, and W. Lee, "Lares: An architecture for secure active monitoring using virtualization," in IEEE Symposium on Security and Privacy, 2008, pp. 233-247.

[25] T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in NDSS, 2003.

[26] VMware. VMware VMsafe. Available: http://www.vmware.com/technicalresources/ security/vmsafe/security_technology.html