1 Zap

1.1 Assumptions and Environment

1. Zap is trying to help legacy applications. How widely are legacy applications used? How legacy applications differ from current application in their requirement for the migrating system, and to other systems? (Guoliang Jin)

2. What assumption did Zap make that controdicts with that of Sprite? (Mengmeng Chen)

1.2 Pods

3. What are the challenges Zap must address to support transparent process migration from one machine to another, regarding resource consistency, conflicts, and dependencies? (Zhenxiao Luo)

4. How does pod avoid creating dependencies among system components that cannot be easily served when a process is migrated? (Mengmeng Chen)

5. Zap only supports adding a process to a pod by creating a new process, not allowing addition of already running processes. Is it possible to modify Zap to support addition of running processes? If so, how; if not, why not? Would such a feature be helpful? (Mark Sieklucki)

6. Zap does not support merging and splitting of pods. The previous question addresses whether merging is possible. How difficult would splitting pods be? Again, is this a feature one that would be useful? (Mark Sieklucki)

7. What the overheads imposed by Zap on processes running outside a pod ? What are the reasons for this overhead on processes outside the pod ? (Leo Prasath)

8. Zap uses the concept of process domain (pod), which is essentially process grouping. What if IPC and process dependencies increase, (which results a large and possibly increasing group size)? Taking it to the extreme, if all the processes in an OS are somewhat related, is it better just to migrate the entire OS? (Yiying Zhang)

9. How could processes communicate using the pod abstraction? What are the benefits, and what are the disadvantages? (Zhenxiao Luo)

10. Zap and Resource Containers (OSDI ’99) follow the same principles, but for different applications. Zap uses pods for decoupling processes from dependencies to the OS and other processes. Resouce containers provide a separate protection domain (resource principal) for a group of processes to accurately account for their resource usage. How can we extend Zap system to provide the full functionality of resource containers (during no-migration periods) and as pods (for migration)? (Mohit Saxena)

1.3 Memory

11. The migration cost of Zap seems higher than VMmigration. Do you think applying pre-copy to Zap could reduce the time cost? Why? (Yupu Zhang)

12. What happens if Zap used pre-copy instead of checkpoint/restore? (Base Paul)

13. Is it possible to implement pod in a live migrated way? Does live migration exclude the use of checkpoint-restart mechanism? (Guoliang Jin)

1.4 Implementation

14. Zap is built as a Linux kernel module. Hence, it supports process migration (pod migration) across machines running independent kernel "images". However, another challenging problem is process migration across machines running independent OS or with independent and distinct architectures. How can the Zap prototype be extended to support this feature, without leveraging a VMM-based solution? (Mohit Saxena)

1.5 Network

15. One feature Zap advertised is its support for networked applications. Imagine there is a web server running in a pod and a normal client (traditional process) is accessing the server from outside the system. What are the techniques Zap is using to support this? How does Zap handle the migration of the server while the client is still accessing it? (Yupu Zhang)

16. The Live migration paper mentioned that Zap did not retain open network connections, while Zap paper says it does (using a proxy for a short duration). Zap's explanation made sense as well. Which paper described it wrong? (Base Paul)

17. Zap did a lot work on network virtualization and migration, while the VMmigration paper pointed out that zap did not address the problem of maintaining open connections for existing services. What is the purpose of the network virtualization and migration in zap? How did VMmigration maintain open connections for existing services? (Guoliang Jin)

18. One of the limitations of VMware (with which the ZAP paper made comparisons in its performance benchmarks) wa the lack of support for migration of networked applications. The live migration paper tackles this problem for Xen VMM. Taking this into account and the pros and cons of the two approaches, discuss which approach is best suited for which scenario (the kinds of applications etc.). (Suhail Shergill)

1.6 Conclusion

19. What are the drawbacks of Zap's approach? (Mengmeng Chen)

2 Live Migration of Virtual Machines

20. Are there instances where live migration of virtual machines is unnecessary or even harmful? If live migration isn't needed, what could be simplified or optimized? (Mark Sieklucki)

2.1 Memory

21. The VMmigration paper uses pre-copy strategy, what should be copied? How should the data be copied? When should stop-copy be used? (Chong Sun)

22. Can the "stunning" mechanism described be improved to allow the process to execute while minimizing impact? Stunning can potentially lead to denial of service. (Varghese Mathew)

2.2 Devices

23. VM migration mechanisms usually don't migrate the state of the device or fail when the source and destination have different network devices (eg. different ethernet cards). Specifically, migrating a live application or VM doing direct I/O to a device on the source to a destination machine may be challenging. How can we address this challenge using the writable working set approach? (Mohit Saxena)

2.3 OS Implementation Issues

24. How is self migration handled in LiveOs migration ? Specifically, how is a consistent OS checkpoint taken while the OS is still running (migrating itself) ? (Leo Prasath)

25. Why are the authors making a big deal with managed vs. self migration. Rather than paravirtualizing, would it not make more sense for the guest to inform the host that I got a migration request, and then let the host do a managed migration? (punting the responsibility) (Varghese Mathew)

26. On the basis of the paper, develop steps required to implement live migration on vmware server or KVM, highlighting the differences applicable. (Varghese Mathew)

3 Comparison

3.1 Assumptions and Environment

27. Discuss the assumptions (regd. the environment parameters etc.) made by each paper and how they impacted their respective scope. (Suhail Shergill)

28. What are the assumptions of environment of two migration systems? What is the difference of design goals? What are the major mechanisms to realize the design goals? (Chong Sun)

3.2 Performance

29. The Live Migration paper keeps the downtime low to the order of milliseconds. What does Xen do to achieve this? Why is it slower in Zap? (Asim Kadav)

30. Both papers talked about the performance of the process of migration. What about the performance in the normal case, i.e. when there's no migration? This would be interesting, since both methods add cost into the normal operation by using some form of virtualization. (Yiying Zhang)

3.3 Complexity

31. Which implementation is more complex? Why is Xen migration much simpler? (Asim Kadav)

3.4 File Systems

32. Can VMmigration use the similar strategy for migrating file systems in ZAP? (Chong Sun)

3.5 Network Issues

33. Which system philosophy could be better scalable to a WAN distributed system? (Jose Angel Perez Rico)

3.6 Other Issues

34. Which system could adapt better to a cluster of heterogenic machines, both in hardware and software? (Jose Angel Perez Rico)

35. Which system could be friendlier to implement, Sprite-like idea of migration to increase parallel execution? (Jose Angel Perez Rico)

3.7 Conclusions

36. What are the pros and cons of Live migrations vs Zap style of checkpoint and restart? (Asim Kadav)

37. What is the advantage of Pod over VMM (running an OS withe same set of processes)? (Base Paul)

38. What are the benefits and drawbacks or limitations of migration at the process level and at the entire (virtual) OS level? What kinds of applications do they fit? (Yiying Zhang)

39. Discuss some implementation decisions that the authors of both papers had to make and their impact (for example, ZAP can handle migration of processes across minor version differences in the kernel because of an implementation decision). (Suhail Shergill)

40. Both Zap and VMmigration use virtual machine techniques to facilitate migration from one machine to another. What are the differences between Zap and VMmigration in terms of using virtualization? What are the pros and cons of each one? (Yupu Zhang)