

### Self-stabilizing Host Systems

### SHOS...

Shlomi Dolev and Reuven Yagel, Computer Science Department Ben-Gurion University of the Negev, Beer-Sheva, Israel

dolev, yagel@cs.bgu.ac.il

Virtualization@SYSTOR 29 Oct. 2007

# Can a guest exhaust a VMM?

- A soft-error moves a KVM guest into VMX-root
  - <u>Not so rare: "a potential error every five hours"</u>
    [<u>Mastipuram & Wee '04]</u>
- A KVM bug?
  - Many lines of new code (> 10KSLOC)...
- Linux security/robustness issues
  - An unaware user runs a faulty program with high privilege
- In our design the host can reach any state and still eventually provide predefined guarantees



### Self-Stabilization

- The combination and type of faults cannot be totally anticipated in on-going systems
- Elegant fault tolerant approach
  - Alternative to the various duplication techniques
  - Started at any state, the system converges to a desired behavior



- "Self-Stabilization in Spite of Distributed Control" [Dijkstra '74]
- Self-Stabilization [Dolev '00]

### A Self-stabilizing Stack



### Self-Stabilizing Operating Systems

### The Problem:

Growing use of autonomous and remote systems (e.g. RPID), but human management is too expensive, nisky or just unavailable, and the combination and type of faulte cannot be totally anticipated in on-going systems (e.g. due to soft errors[b])



### Event: Remote Space Vehicle Failure 181

The Solid owner has a posision bendered \$6000 /\$11 from Lockberd, Martin The approximate instanton-barranees because to those the definition of the second s Spirit fell silent, clone on the emptiness of More.

### Proposed Solution:

-To build on the well designed and well underwised paradigm of self-stabilization usua on me wei oraspece and weil usdawindo proclam of welf-etabilize (which traditionally is being used in distributed systems)
 Thereby achieving transferme, etc.
 Using welf-stabilization:

-A system can be started in an arbitrary state and converge to a desired behavior, thus,

-Following any sequence of transient faults, the (operating) system converges -Self-stabilizing algorithms cannot be run unless hardware-OS are stabilizing (by use of "fair composition" [2])

Wain opproaches: -Black box: adding monitoring layer to an existing operating system

-Tailored: building a (tiny) kernel with basic OS functions, such as pro-scheduling, memory & IO devices management

### Assumptions:

Whole soft-state can be corrupted (including e.g. Program Counter) -Wicroprocessor is self-stabilizing [3]

A quote from Intel® Pertium nonual [7] demonstrates that the processor can reach states in which no self stabilizing program can execute: ", if the ESP or SP register is 1 when the PUSH instruction is executed, the processor shuft down,"

### Example: Memory Management [5]

-Added requirements -Eventual Consistency of various levels of the memory hierarchy, e.g. RAM and Hard-disk

RAW and Hard-disk -Eventually Self-stabilization preservation of processes, in spite of sharing of the memory nexurces -Three scaled solutions, demonstrating:

-Pull swopping -Pixed partitioning -Dynamic allocations with con-

References: [1] E. W. Dijkstra, "Self-Stabilization in Spite of Diffebuted Control", Communications of the ACM, Vol. 17, No. 11,, 1974 [2] S. Doley, Self-Stabilization The MIT Press, 2000. [3] S. Delex, Y. Horix, "Self-Stabilizing Microprocessor, Analyzing and Overcoming Soft-Errors", 17n International Conference on Architecture of Computing Systems 39, 31-46. 2004

[4] S. Delex, R. Yagel, "Towards Self-Stabilizing Operating Systems", 2nd International Workshop on Self-Adaptive and Autonomic Computing Systems - DEXA, pp.684-688, 2004. [5] S. Dolev, R. Yagel, "Memory Hanagement for Self-Stabilizing Operating Systems". To appear in Proceedings of the 7<sup>th</sup> Int. Symposium on Self Stabilizing Systems, 2006. [6] M. Kintler et. d. "Modeling the effect of technolo friends on the soft error role of combinational logic". *ICDSN*, volume 72 of LNCS, pages 216--226, 2002. [7] http://developer.intel.com/design/pentium4 [8] http://www.estimes.com/story/OE6200402205 9 http://www.cs.bgu.cc.il/~yogel/sos



Self-stabilization self-stabilizing system is a system that can automatically secover following the occurrence of (transient) foults [12]

### Solution Foundations (4): Satisfying program loading & process scheduling by:

-Partions of code in ROM -Really Non-Maskable Internupt and Watchdog architecture -Periodic reset reinstall & execute, or -Continuous monitoring and consistency enforcement of the whole motion state. by the scheduling algorithm

### Method:

Define additional requirements for each main OS Processor (e.g. Pentium [6]) instruction manual defines a transition function

 Gradually evolve simple self-stabilizing solutions that also follow computer-architecture\OS progress
 Built on previous stages Detailed proof for self-stabilization of algorithms AND implementation

Consistency achieved through continuous checks and consistency establishment of data structures Stabilization preserving via



### Conclusions:

The work shows theoretical and practical ways to achieve the goal of a self-stabilizing operating system -Proved & verified prototype implementations of SOS are available [9]. 4

### Stabilizing Reputation & Trust

- Give a chance to change reputation, both ways...
- Maybe the reputation history is corrupted
- Constantly fading old reputation and accumulating new reputation
- Thus, stabilization of reputation and trust



# Main SHOS Concepts

- Secure booting of minimal TCB
- Offline Byzantine behavior detectors
- Runtime anti-Byzantine contract enforcers
- Stabilizing trust and reputation
- Self-stabilization
- Stability of non-Byzantine programs is preserved from any state



### Conclusion





- We presented a general design for stabilizing host systems
- Virtualization can benefit from the foundation of self-stabilization

http://www.cs.bgu.ac.il/~yagel/sos