http://ssrg.ece.vt.edu

Systems Software Research Group

ece Bradley Department of Electrical & Computer Engineering

VirginiaTech Invent the Future®

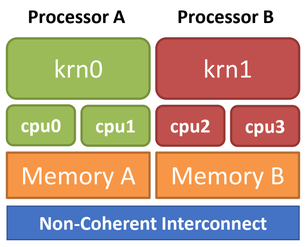
A Page Coherency Protocol for Popcorn Replicated-kernel Operating System

Marina Sadini, Antonio Barbalace, Binoy Ravindran, Francesco Quaglia*
Department of Electrical and Computer Engineering, Virginia Tech, Virginia, USA
{sadini, antoniob, binoy@vt.edu}
*Francesco Quaglia is with DIAG, Sapienza, University of Rome, Italy {quaglia@dis.uniroma1.it}

Future computing platforms are increasingly ISA-diverse and parallel

Scenario

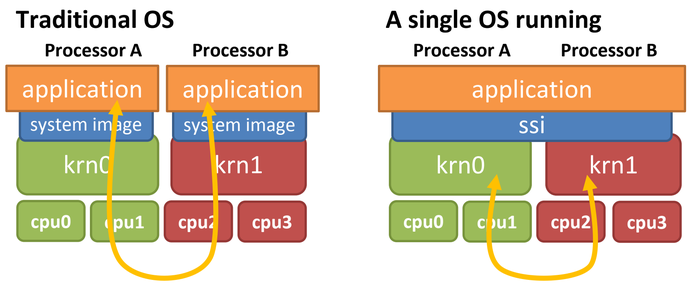
- Different interconnected processors (or machines)
- Each has its own (private) memory



Goals

- Illusion of a single OS running (like in SMP setups)
- Applications do not need to be rewritten/recompiled

- Explicit communication in the application (e.g. MPI)



- Kernels communicate instead of the application (i.e. DVSM)

Popcorn Linux

- A replicated-kernel OS
  - Similar in concept to a multikernel (i.e. Barrelfish) but ..
  - Based on Linux 3.2.14
- Open-source and freely available
- Can be used to simulate a Het. setup on Homogeneous hardware

Inter-Kernel Communication

- Communication take place by messages
- The messaging is built on top of a combination of shared memory buffering and IPI notification
- Mixture of interrupts and polling
- The multicast/broadcast is not scalable (no x2APIC)

Inter-Kernel Thread Migration

- When a thread migrates from kernel A to kernel B
- **If** other threads of that process reside on kernel B
  - **remote thread cloning**
- **Else if** it is the only thread of that process on kernel B
  - **remote thread creation**

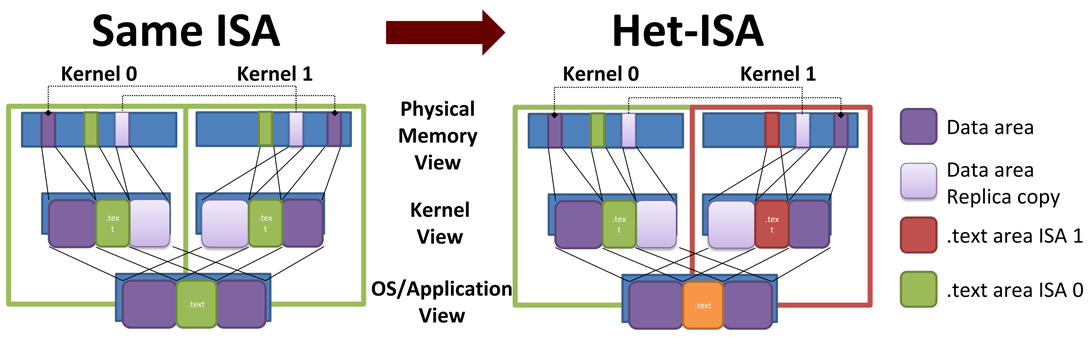
Inter-kernel Thread Migration, benefits:

+ Single System Image

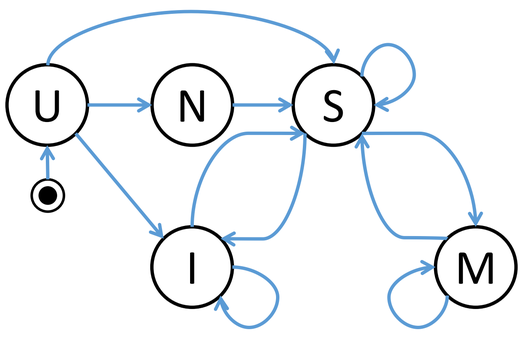
Load balancing, power saving, memory/device locality, performance, expanded memory space, greater device count, etc.

Address Space Page Replication and Coherency to deploy thread migration

- Replicated page table per kernel (same application)
- Replicated page content on each kernel
- Page level protocol to guarantee a coherent memory view to the app.



New Page Coherency Protocol



- We started from the MSI coherency protocol extending it with *Unmapped* and *NotReplicated* states
- Linux pages are not present at initialization (*Unmapped*)
- On-demand pages become present and access any of the other states (N, M, S, I)

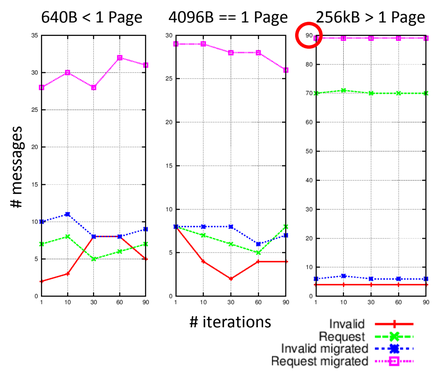
Why a new protocol?

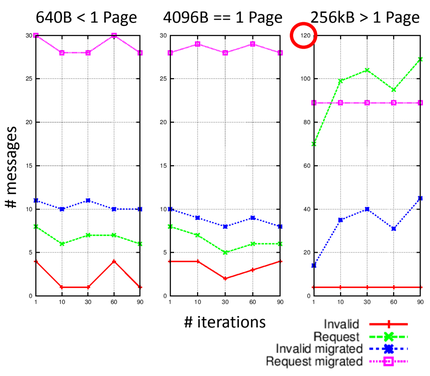
- A distributed algorithm with voting (i.e. no centralized approach)
- Less constraints on the hardware (e.g. no global clock)
- Designed for heterogeneous platforms

Protocol's Initial Evaluation

R,R Workload

- Increasing the number of iterations does not change the number of page requests
- All pages are locally copied during the first iteration
- No more requests are generated in the successive iterations



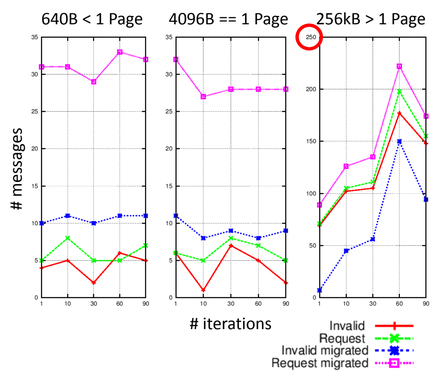


R,W Workload

- The number of messages required to update the reader increased
- All the other messages are kept around the same value
- The communication overhead only slightly increases (<20% total)

W,W Workload

- Increasing the number of iterations does change the number of messages
- Both replicas are concurrently updating the array invalidating the content of the pages
- Requests are generated during each iteration



Conclusions

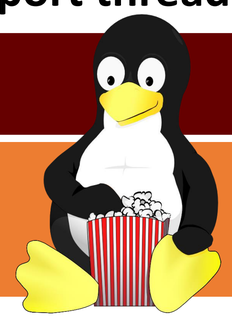
We designed, implemented and tested a new page coherency protocol in Popcorn Linux.

The design properties are satisfied and the implementation is effective. The voting scheme is a promising solution.

New software components have been added to Popcorn sources (and Linux) to support thread migration (i.e. SIGFORK).

www.popcornlinux.org

ACKNOWLEDGMENTS
This work is supported by US Office of Naval Research under Contract N00014-12-1-0880.



DEPARTMENT OF THE NAVY ONR Science & Technology