PROBLEM STATEMENT
A global coherent address space becomes infeasible at cloud or datacenter scales and partitioning the address space does not achieve elasticity for all applications. We focus on:
- In-memory databases for distributed web-scale applications
- Graph processing and search algorithms
- Applications too complex to redesign for a distributed model

ELASTIC OPERATIONS

STRETCHING
A process experiencing signs of contention such as thrashing can be stretched across multiple nodes. A “ghost” process is created on a remote node enabling pages to be moved in and out of the new process’s address space.

We implemented stretching in the Linux 3.6.11 kernel based on the Kerrighed SSI scheme.

glNUMA
(generalized-locality NUMA)
The Linux scheduler has inadequate information on memory locality for scheduling a stretched process. Multi-level or asymmetric NUMA architectures also need a better representation of locality. glNUMA is a Linux 3.18 kernel modification that:
1) Allows an accurate representation of NUMA memory latencies to be used by the Linux kernel
2) Uses a weighted sum measure of per-NUMA-node page access counts to schedule tasks

On a 4 processor NUMA system running the PARSEC benchmark fluidanimate, glNUMA resulted in 46% fewer process migrations and decreased runtime by 6.7%.

OUR VISION
ElasticOS: an alternative to coherent single-system images: Leverage hardware cache coherency where it is present, but enable new elastic operations for maintaining the single address space abstraction across a datacenter.

JUMPING
Once a process has been stretched, jumping can be triggered by the ElasticOS manager in two cases:
1. Lightweight (common): The ElasticOS manager detects excessive pulls from a remote node. On the next remote page fault a jump is triggered instead of a pull.
2. Heavyweight: The process is experiencing slowdowns from resource contention and an underutilized node exists. The ElasticOS manager can trigger a jump, bringing along data pages (process migration).

PULLING/PUSHING
Page faults can be serviced by remote memory. A remote memory page is pulled after a configurable number of consecutive accesses and the original copy is invalidated. Pulling transfers ownership of a page so cache coherency traffic is avoided. Likewise, a page can be pushed to a remote node after stretching.

FUTURE WORK
- Develop the ElasticOS userspace manager with: the ability to coordinate among distributed managers, and a resource allocation strategy using the constraints of memory latency and network contention
- Improve jumping implementation to allow migration in both directions
- Evaluate glNUMA behavior on a multi-level NUMA topology (more than 4 processors)

ACKNOWLEDGEMENT
This material is based upon work supported by the National Science Foundation under Grant No. (1337399). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.