Virtualization, Empathic Systems, and Sensors
Current Work in the Prescience Lab

McCormick

Peter A. Dinda
presciencelab.org
Electrical Engineering
and Computer Science
Northwestern Engineering

Virtualization (v3vee.org)

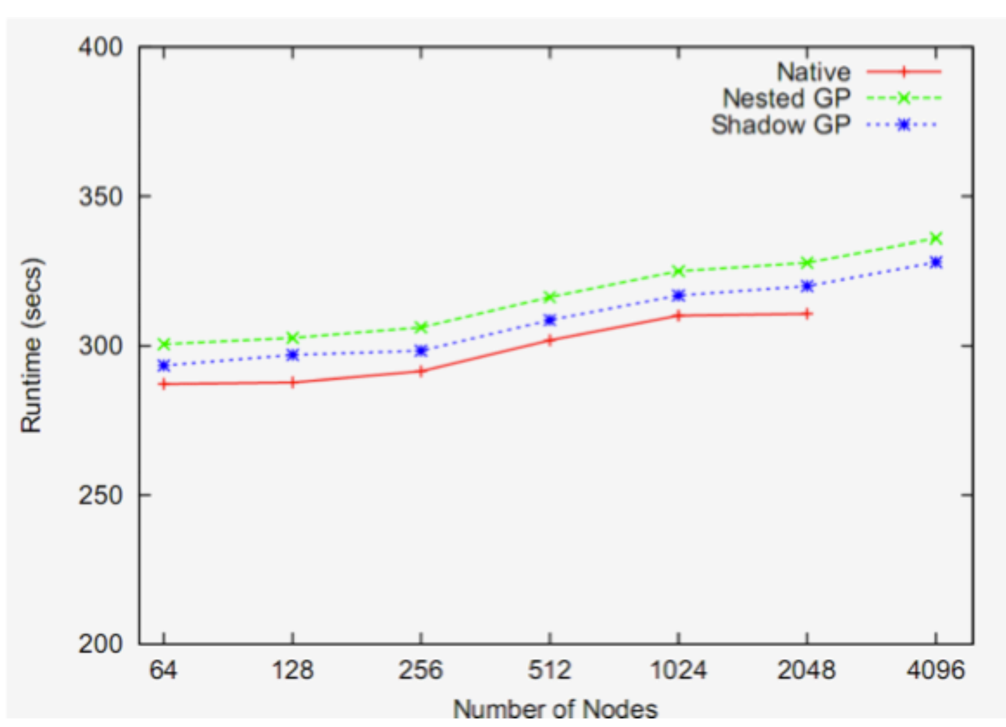
Palacios
An OS Independent Embeddable VMM

V3VEE

A new, publicly available, BSD-licensed, open source virtual machine monitor for modern x86 architectures that runs on Cray XT supercomputers, clusters (Infiniband and Ethernet), servers, desktops, etc.

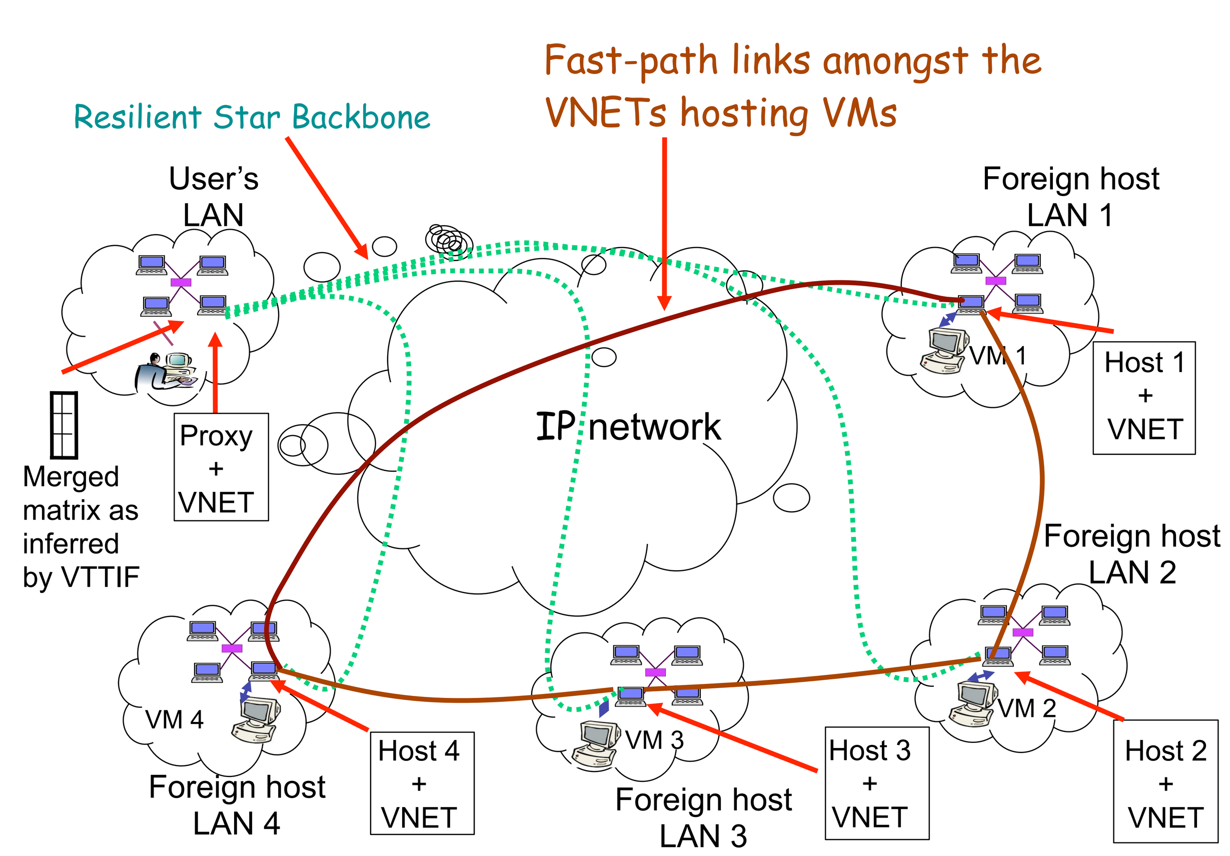
Palacios is intended to support research in high performance computing and computer architecture, in addition to systems. It can be easily embedded into other OSes. Current embeddings include Linux, Kitten, and Minix 3.

Palacios, when embedded in a lightweight kernel, such as Sandia National Lab's Kitten, forms a compact, type-I pure VMM suitable for virtualizing a supercomputer at scale with minimal overhead even when running tightly coupled, communication-intensive parallel applications on HPC OSes.



Virtuoso: Adaptive Virtualized IaaS Cloud Computing
virtuoso.cs.northwestern.edu

- Providers sell computational and communication bandwidth
- Users run collections of virtual machines (VMs) that are interconnected by overlay networks
- Replacement for buying machines
- Continuously adapts to increase the performance of your existing, unmodified applications and operating systems

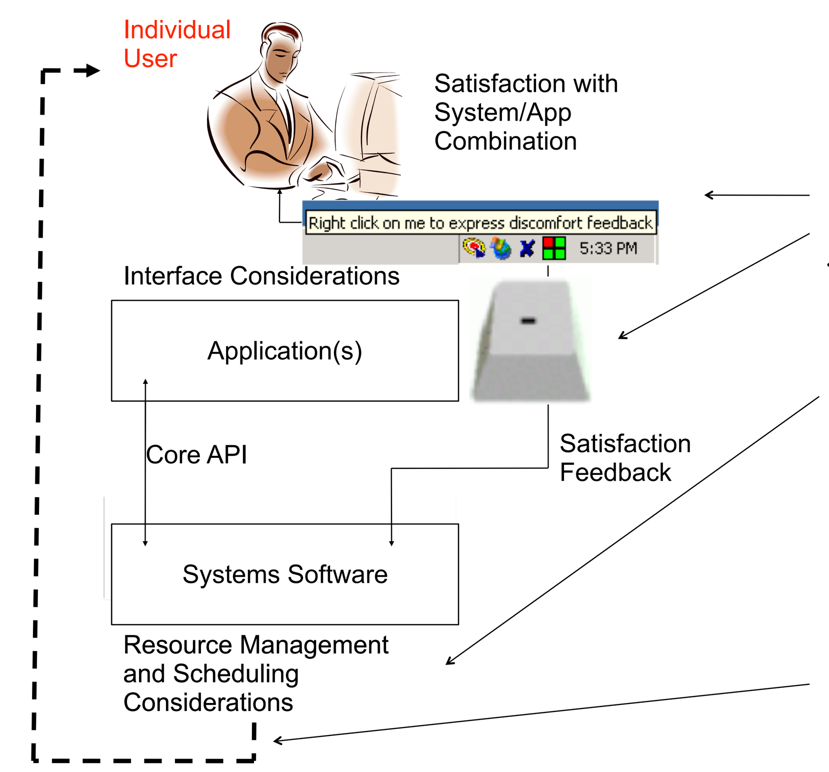


- Monitor application traffic: Use application's own traffic to automatically and cheaply produce a view of the application's network and CPU demands and of parallel load imbalance
- Monitor physical network using the application's own traffic to automatically and cheaply probe it, and then use the probes to produce characterizations
- Formalize performance optimization problems in clean, simple ways that facilitate understanding their asymptotic difficulty
- Adapt the application to the network to make it run faster or more cost-effectively with algorithms that make use of monitoring information to drive mechanisms like VM->host mapping, scheduling of VMs, overlay network topology and routing, etc.
- Adapt the network to the application through automatic reservations of CPU (incl. gang scheduling) and optical net paths
- Transparently add network services to unmodified applications and OSes to fix design problems

Empathic Systems (empathicsystems.org)

Experimental Computer **Systems** Researchers Should…

- Incorporate **user studies** into the evaluation of systems
  - No such thing as the typical user
  - Variation in user satisfaction with given operating point is huge
- Incorporate **direct user feedback** into the design of systems
  - No such thing as the typical user
  - Measure and leverage that high user variation



Human Interface to the Systems Software, not just the Application…

…to control systems-level decision making that impacts the user experience…

…via global feedback control that incorporates the user

We Have Applied This Idea Extensively and Successfully

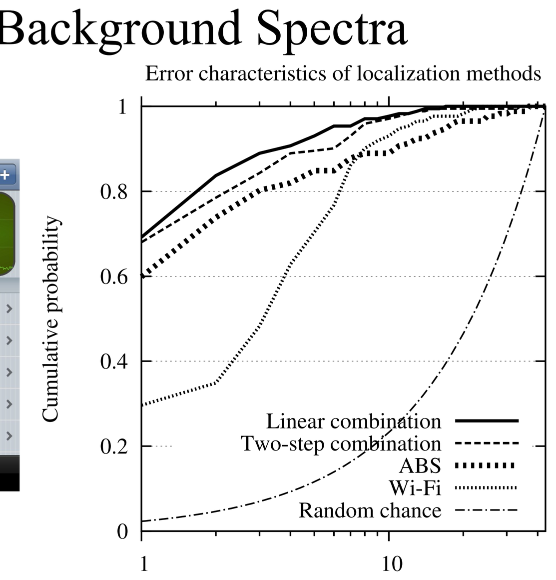
- **User-driven scheduling of interactive virtual machines** allows even naïve users to trade off between cost and interactive performance via a simple tactile interface
- **User-driven dynamic voltage and frequency scaling** exploits user feedback to lower power consumption on a laptop computer by considerable amounts while maintaining high user satisfaction.
  - UDFS (user presses button when irritated): 22% better than Windows DVFS
  - PICSEL (evaluates rate of screen content change): 12.1% better
  - iDVFS (neural net maps from hardware measures to per-user satisfaction): 25% better
  - PTP (biometrics-based satisfaction): 12% better
- **Speculative remote display** predicts and draws server screen content on the client, ameliorating network latency effects. Naïve users can trade off between display correctness and responsiveness.
- **Empathic network link scheduling** provides user satisfaction-driven control over scheduling the broadband router link in home networks. It increases overall user satisfaction by 24% over an FCFS link, and by 19% compared to a static WFQ link.
- **User-presence-driven display power management** controls laptop LCD backlight based on presence determined by ultrasonic sonar on commodity hardware

We Are Now Studying Techniques for Free, Biometric-based, Continuous Measurement of User Satisfaction

Left to right: keystroke force sensors, galvanic skin response, pupil dilation, active ultrasonic sonar
Not shown: blood pressure, pulse rate and dynamics, temperature, infrared camera, accelerometer

Room-level Localization using Acoustic Background Spectra

- Determine room based on acoustic signature
- **No infrastructure required**
- New rank-order filtered repeated spectral analysis technique can be implemented on mobile phones
- Technique can be combined with wifi localization to enhance accuracy



Try it now!
"BatPhone" on Apple App Store

Sensors (absynth-project.org)

Wireless sensor network applications are extremely challenging for domain scientists to implement.

Success typically requires either collaboration with a "CS side" sensor networking researcher or with an expensive embedded systems engineer.

However, many prospective applications are either conceptually simple or fit into one of a small number of classes.

We design, implement, and evaluate (through carefully controlled user studies) programming languages and systems specifically for domain scientists and other non experts

Archetype-based Design

- Study of the literature for deployed applications suggests almost all fit into seven classes.
- Proposal: develop an "archetype language" for each class combined with an generic template (an "archetype") in that language. The user answers questions about their potential application to lead to an archetype. He modifies the archetype for his specific purposes. The system synthesizes a hardware/software design.
- Archetype languages are designed for domain scientists. They are also extremely high-level, allowing expression of the archetype in a page of code, and freeing the hands of the synthesis and compilation toolchains.

Proposed language for first identified archetype has high success rate and low development time in user study comparing it to other languages →

Language	Success rate			Develop time (min)		
	T1	T2	T3	T1	T2	T3
SwissQM	3/3	3/3	N.A.*	5.7	11.3	N.A.
WASP	2/2	2/4	2/4	16	31	29.5
TinySQL	3/4	2/3	0/3	17.7	27.5	N.A.
TinyTemplate	1/4	0/3	0/3	34	N.A.	N.A.
TinyScript	0/3	0/3	0/4	N.A.	N.A.	N.A.
WASP2	3/3	3/4	2/3	3	9.7	23.5

A BASIC Approach

- The BASIC programming language proved to be a great success in getting naïve users (children) to write simple programs on resource-constrained embedded systems (the millions of home computers of the early '80s).
- We have developed an BASIC for use in sensor networks. The language is extended with sensor network concepts needed for writing node-oriented programs, and these concepts are presented via user study-tested constructs found to be sensible to non-programmers.
- Depending on the task, 45-55% of subjects with **no** prior programming experience can write simple, power-efficient, node-oriented sensor network programs after a 30 minute tutorial. 67-100% of those matched to typical domain scientist expertise can do so.

Peter Dinda is a professor in the Department of Electrical Engineering and Computer Science at Northwestern University, and head of its Computer Engineering and Systems division, which includes 17 faculty members. He holds a B.S. in electrical and computer engineering from the University of Wisconsin and a Ph.D. in computer science from Carnegie Mellon University. He works in experimental computer systems, particularly parallel and distributed systems. His research currently involves virtualization for distributed and parallel computing, programming languages for parallel computing, programming languages for sensor networks, and empathic systems for bridging individual user satisfaction and systems-level decision-making.

Collaborators on the efforts noted here are Jack Lange (U. Pittsburgh), Patrick Bridges (U. New Mexico), Kevin Pedretti (Sandia National Labs), Gokhan Memik (Northwestern), Robert Dick (U. Michigan), and our amazing students.

NSF
Sandia National Laboratories
Oak Ridge National Laboratory
symantec.
Dell
vmware
Microsoft