SAND2009-4763C

OVIS: Scalable Run Time Data Collection, Analysis, and Visualization

OVIS Team:

Jim Brandt, **Frank Chen**, **Ananya Das**, Vince DeSapio, Ann Gentile, Jackson Mayo, Philippe Pébay, Diana Roe, Don Rudish, David Thompson, and Matthew Wong

http://ovis.ca.sandia.gov

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

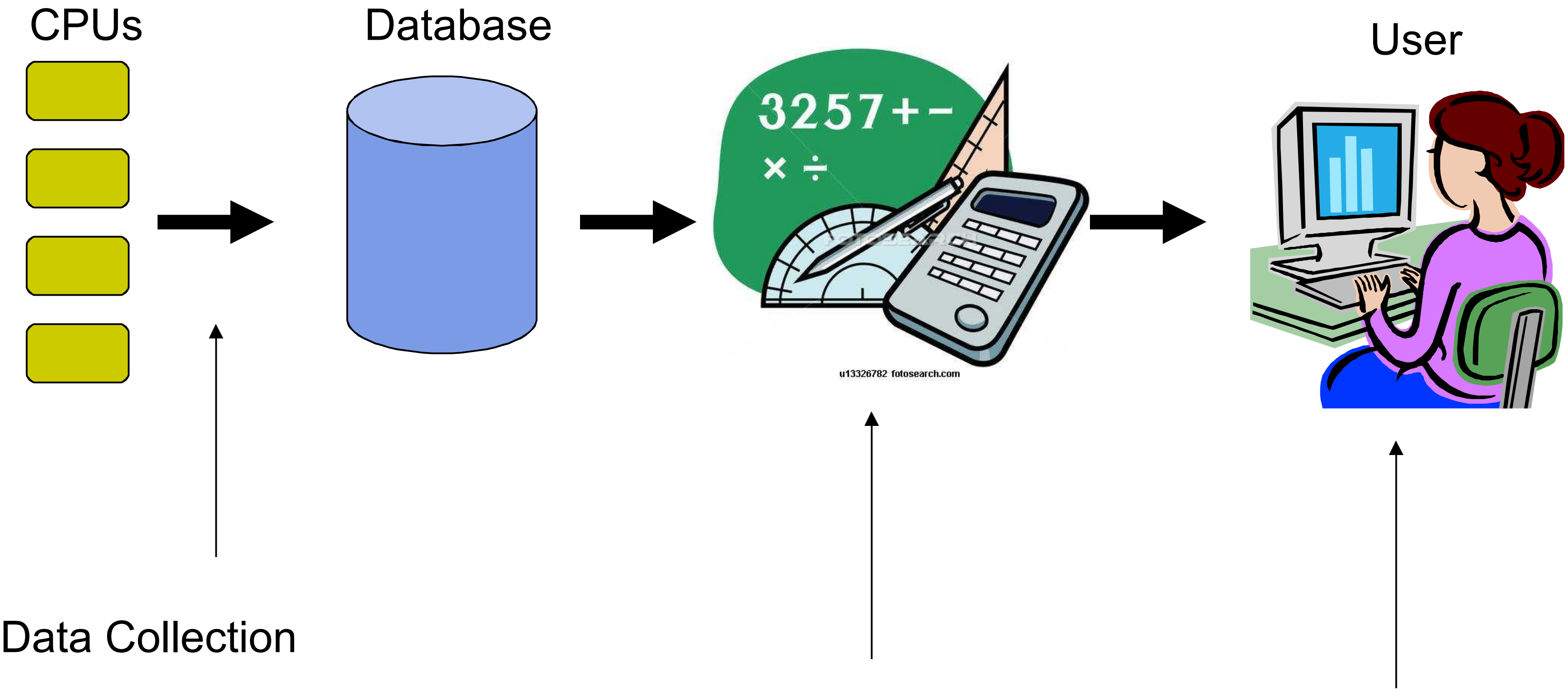
Talk Contents

- Project / Architectural Overview
- High Performance Computing Applications
  - Demo
- Network Applications
  - Demo

OVIS Project Overview

- **Easy to Use Graphical Application**
- **Large Scale Data Collection and Visualization**
  - Millions of Components
  - ~100 metrics per component
  - Sample rate 20 per minute
  - Run time and post run analysis
  - Run time and post run visualization
- **Applications**:
  - High Performance Computing (HPC) Reliability, Availability, Serviceability (RAS) Systems
  - Networked Environments

Architectural Overview



CPUs

Database

User

Data Collection

Data Storage: Creating & Loading Database

Mathematical Analysis

Human Interaction

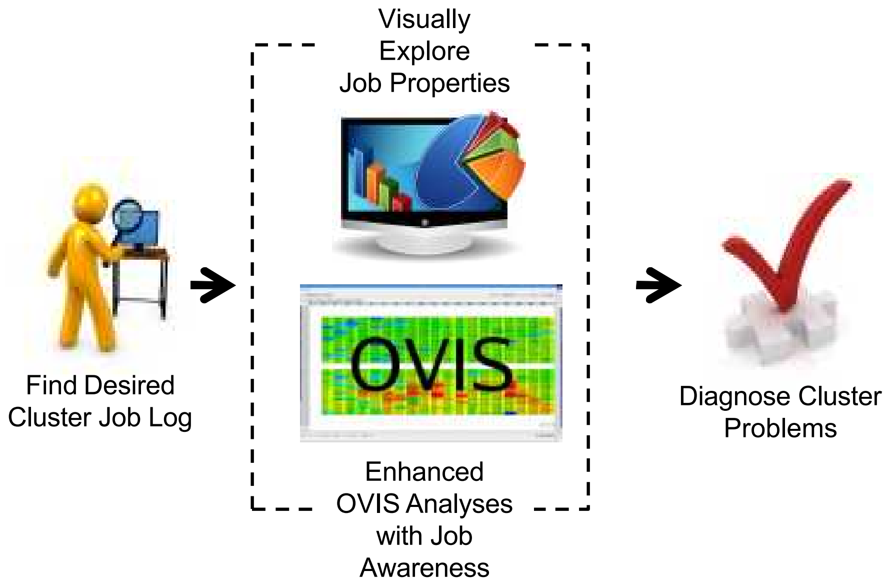
HPC Applications

- Motivation:
  - Larger platforms are harder to troubleshoot
- System Administrator Needs:
  - Identify system failures as soon as possible
  - Determine causes of failures, faulty components
  - Collect data to Predict / Prevent future failures
    - More Resiliency, Availability, and Serviceability

The OVIS Approach

- Scalable Data Collection
- Analysis algorithms for Failure Prediction
  - Anomaly Detection: Correlation of low probability events
  - Determine Faulty Components
  - Predict Future Failures
  - Measurement of Confidence for failure prediction
- User Interface / Visualization

The OVIS Approach



Find Desired Cluster Job Log

Visually Explore Job Properties

Enhanced OVIS Analyses with Job Awareness

Diagnose Cluster Problems

Visualization Enhancements:

- Pie Chart and Summary Views
- Job List Node Coloring on 3D View

Usability Enhancements:

- Enhanced Parser
- Subselections in Pie Chart & Summary View

Network Systems: HPC and Security

- Apply Large Scale Fault Detection to Networks:
  - Local Sandia Networks
  - Government Networks
  - Controlled Systems → Large Scale Systems
- Motivations:
  - Reliability, Availability, and Serviceability
    - Increase Network Resiliency
  - Network Security
    - Defend against attacks

HPC Networks and Network Security: Approach

- Real-Time Analysis
- Determine current network connections
  - Large-scale data collection
- Detect anomalies:
  - Abnormal traffic to/from a node
  - Abnormal traffic on a link
  - Deviations from observed time-based patterns
  - Deviations from observed event-based patterns
- User Interface allows easy interaction between users and data

Conclusions & Future Work

- OVIS provides a graphical interface between users and large-scale data
  - Visual exploration
  - Correlative Analyses
  - Anomaly detection
- Web search interface, GNUPlot integration, Sys Admin Job Analyzer
- Local network approaches may be applicable in large networking environments
- Open Source OVIS: http://ovis.ca.sandia.gov

Questions?

Open source OVIS: http://ovis.ca.sandia.gov

Thank you!

- Any questions?
- We’ll have a poster!

Demo of Cyber Emulation

Overview of HPC Applications

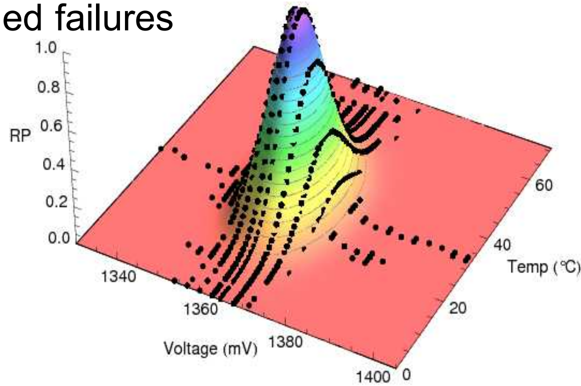
- **Motivation**
  - As platforms grow in size and complexity administrators are increasingly dependent on home grown scripts and knowledge bases to troubleshoot problems.
  - Knowledge transfer is difficult and each new system takes more time to get familiar with
- **Tools**
  - Identify failure as soon as possible
    - Remove failed resource from available resource pool
    - Re-run application with replacement resource
    - Release unused resources tied up by hung application back into resource pool
  - Figure out cause of failure that occurred so that it can be repaired and returned to service ASAP
    - Enter symptoms and related cause into appropriate knowledge base
  - Assist users in troubleshooting when applications aren’t behaving as expected
    - Understand relationships between placement and performance

Resilience Through Fault Prediction

- Accurate prediction of faults with time to respond
  - Target checkpoint and/or migration only on/from failing component(s) and only when failure is imminent
    - Decreased time to completion
    - Increased effective utilization of platform
- Accurate root cause analysis
  - Target failing and all potentially affected components

Resilience Fault Prediction Strategy

- Discover predictors, accuracy, time windows, and coverage with respect to all non-recoverable faults
  - Scalable data collection
    - HW related metrics
      - Limited by current instrumentation
      - Discovery can help drive future system instrumentation
    - System related metrics
      - RM databases, log files, troubleshooting notes, etc.
      - Work with System Administrators to capture as much as possible
    - Not available
      - Human errors, power grid outages, etc.
  - Scalable data analysis
    - Definition of analysis methods that make sense given the data and time scales
      - Currently: correlate low probability behaviors with logged failures
    - Efficient data exploration tools
      - UI
      - Visualization
- Quantify Prediction Effectiveness



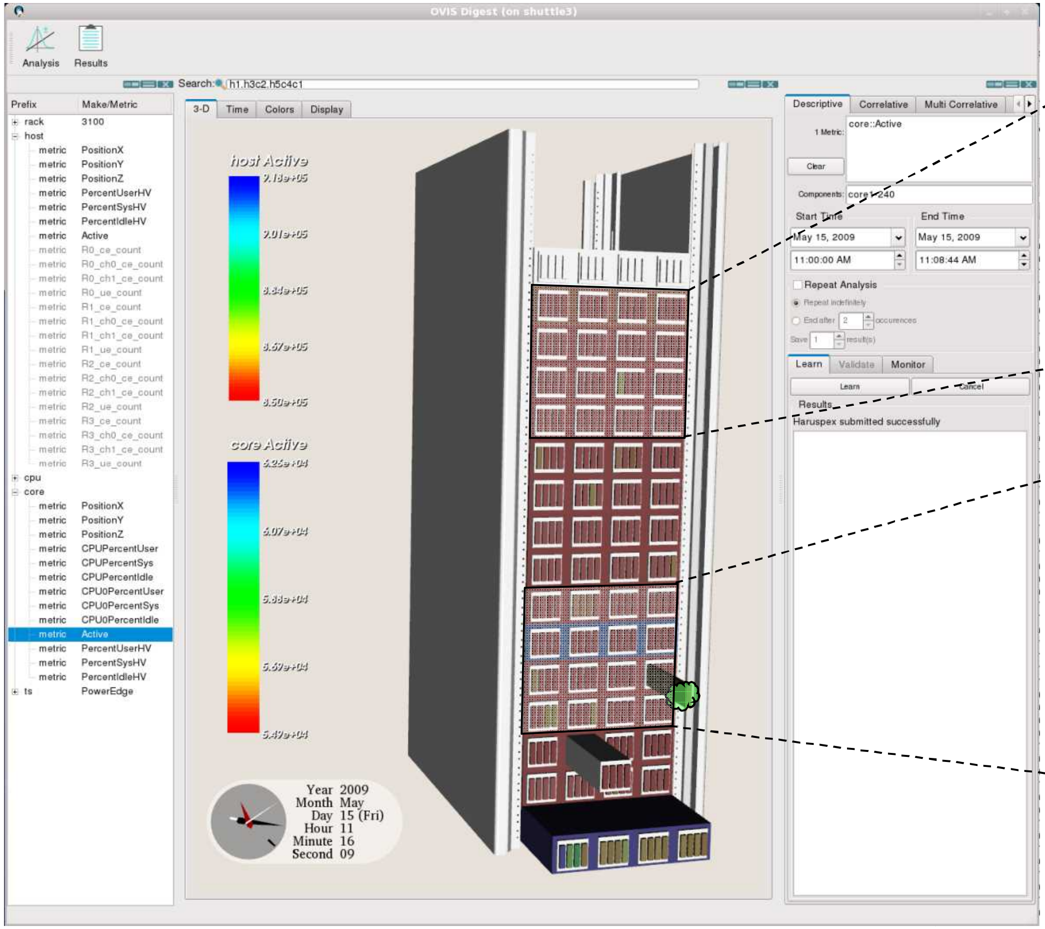
Resilience Status

- Scalable system designed and implemented
  - Data collection, analysis algorithms, UI, visualization
- Deployed on TLCC cluster (GLORY)
  - Three main failure modes
    - Out of Memory, Stuck CPU, Power Supply
- Results
  - Out of Memory failure precursor discovered
  - Have implemented additional information collection for exploring Stuck CPU precursors
  - Detection of power supply failure precursors will require additional instrumentation
- Effectiveness scoring algorithm defined for quantifying our ability to predict failure

Resilience Future

- Enhancements to UI tools for ease of exploration
- Addition of analysis tools
  - Temporal behaviors of ensembles of metrics
  - Automated correlations between metrics states and behaviors with identified faults and failures in logs and RM databases
- Enhancements to visualization to facilitate understanding
- Additional instrumentation
  - HW through vendor interaction
  - SW through log file parsers and additional collection of system state
- Deployment on more platforms
  - Red Sky
  - Other current generation TLCC (Whitney and perhaps LLNL)
  - Next generation TLCC
  - CRAY XT5 and beyond?

OVIS Physical Display of Whitney Testbed
16 nodes, 64 CPUs, 256 cores



OVIS Digest (on shuttle3)

Search: h1.h3c2.h5c4c1

Prefix	Make/Metric
rack	3100
host	
metric	PositionX
metric	PositionY
metric	PositionZ
metric	PercentUserHV
metric	PercentSysHV
metric	PercentIdleHV
metric	Active
metric	R0_ce_count
metric	R0_ch0_ce_count
metric	R0_ch1_ce_count
metric	R0_ue_count
metric	R1_ce_count
metric	R1_ch0_ce_count
metric	R1_ch1_ce_count
metric	R1_ue_count
metric	R2_ce_count
metric	R2_ch0_ce_count
metric	R2_ch1_ce_count
metric	R2_ue_count
metric	R3_ce_count
metric	R3_ch0_ce_count
metric	R3_ch1_ce_count
metric	R3_ue_count
cpu	
core	
metric	PositionX
metric	PositionY
metric	PositionZ
metric	CPUPercentUser
metric	CPUPercentSys
metric	CPUPercentIdle
metric	CPU0PercentUser
metric	CPU0PercentSys
metric	CPU0PercentIdle
metric	Active
metric	PercentUserHV
metric	PercentSysHV
metric	PercentIdleHV
ts	PowerEdge

Year 2009
Month May
Day 15 (Fri)
Hour 11
Minute 16
Second 09

1 Metric: core::Active

Components: core1-240

Start Time: May 15, 2009 11:00:00 AM

End Time: May 15, 2009 11:08:44 AM

□ Repeat Analysis

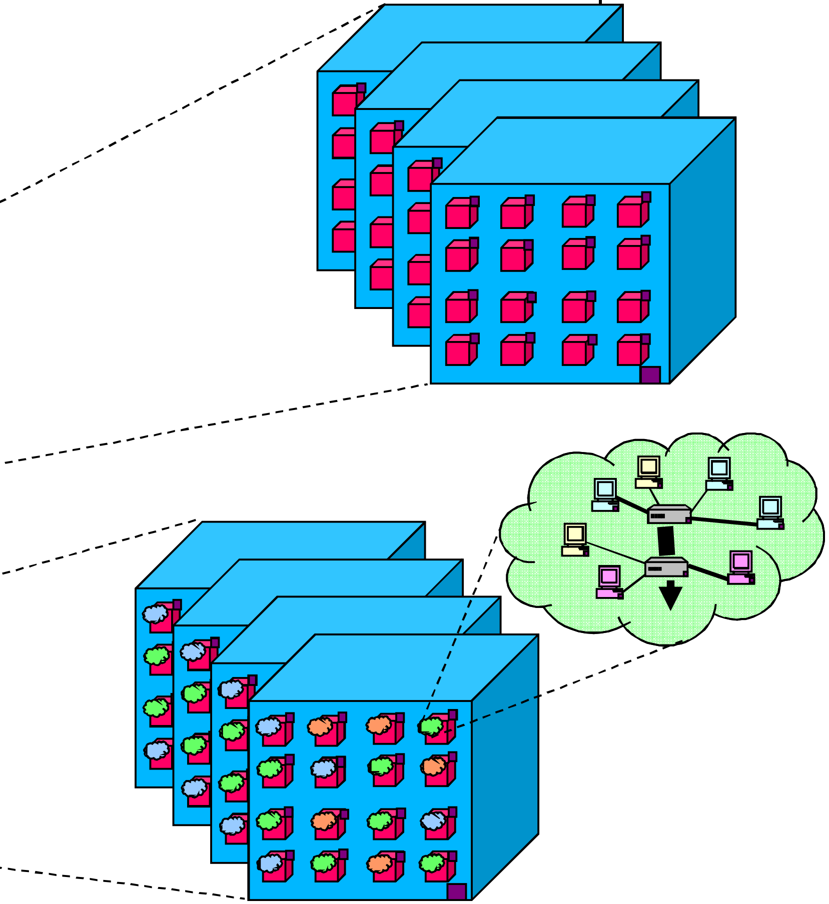
● Repeat indefinitely

○ End after 2 occurences

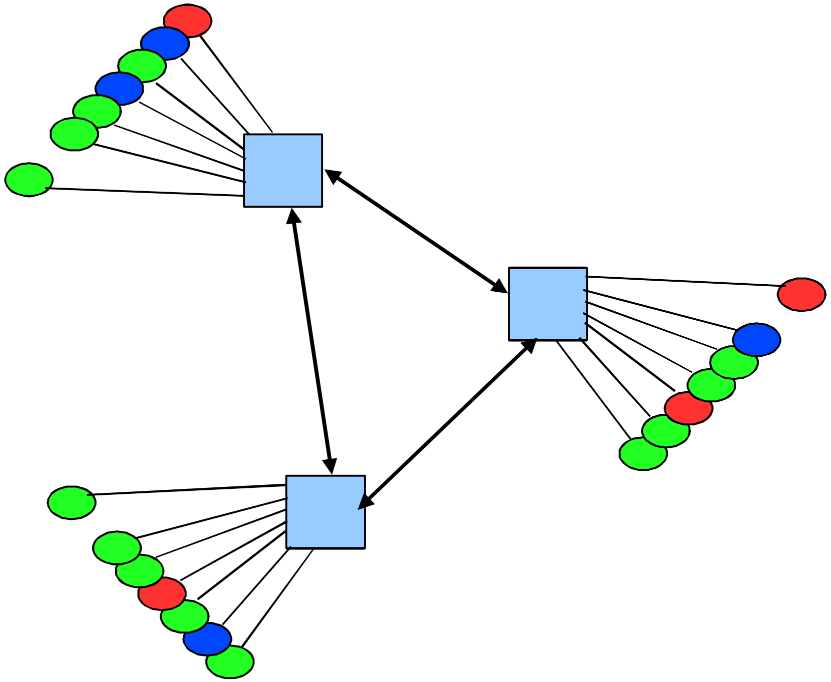
Save 1 result(s)

Results

Haruspex submitted successfully



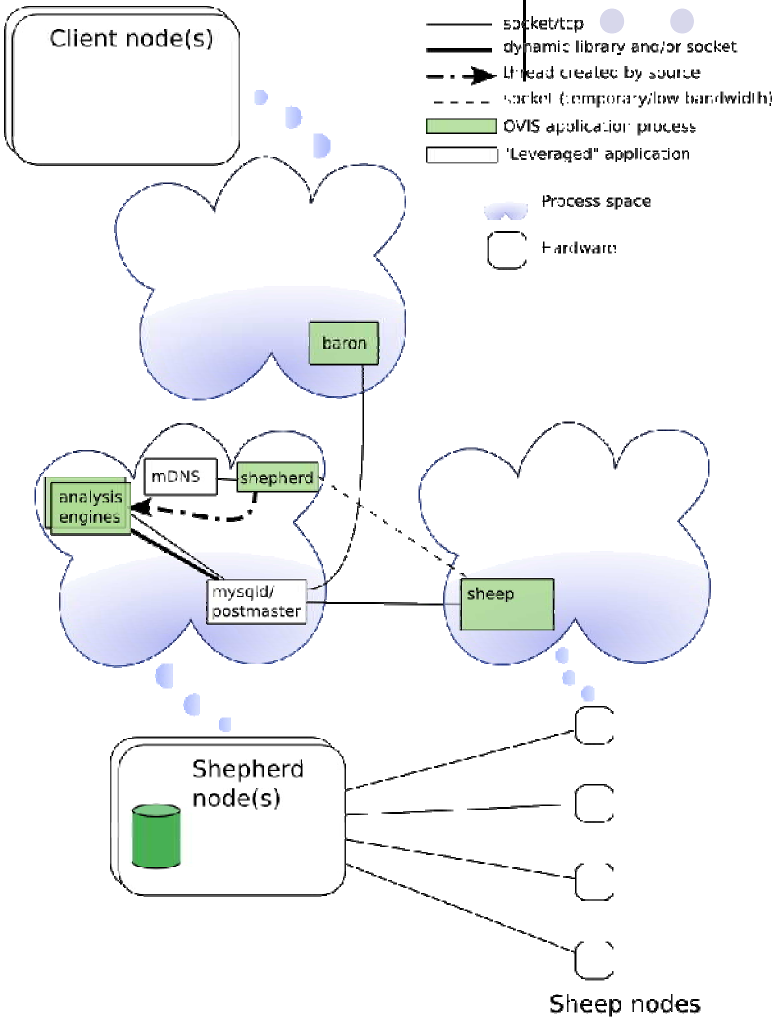
Attack Simulation and Analysis



- 1 VM per core running full OS
- Multiple address spaces
  - VM's routing
- •Simulating Heterogeneous Entities under attack
  - App running in each VM
  - Non-uniformly susceptible to attack
- •Attack
  - Detectable effect on entity (CPU Utilization change)
  - Successfully attacked entities propagate the attack
- •Run-time monitoring and visualization of attack effects and propagation
  - Data collectors running in each VM
  - App CPU Utilization change
  - Attack metric

Architectural Overview

- OVIS is a suite of 3 applications – baron, shepherd, sheep – sharing a common database schema (more on next slide)
- **Database Digest**: VTK/Qt User interface to Database
  - Used by RAS/statistics researchers and system admins
  - Separates view of the same set of components into panels
- **Service-node program**:
  - Advertises DB availability
  - Responds to requests for analyses (haruspices)
- **Service-node / compute-node program**:
  - Listens for shepherds
  - Stores measurements to database on shepherd node



socket/tcp
dynamic library and/or socket
thread created by source
socket (temporary/low bandwidth)
OVIS application process
"Leveraged" application
Process space
Hardware
Client node(s)
baron
mDNS
shepherd
analysis engines
mysqld/ postmaster
sheep
Shepherd node(s)
Sheep nodes