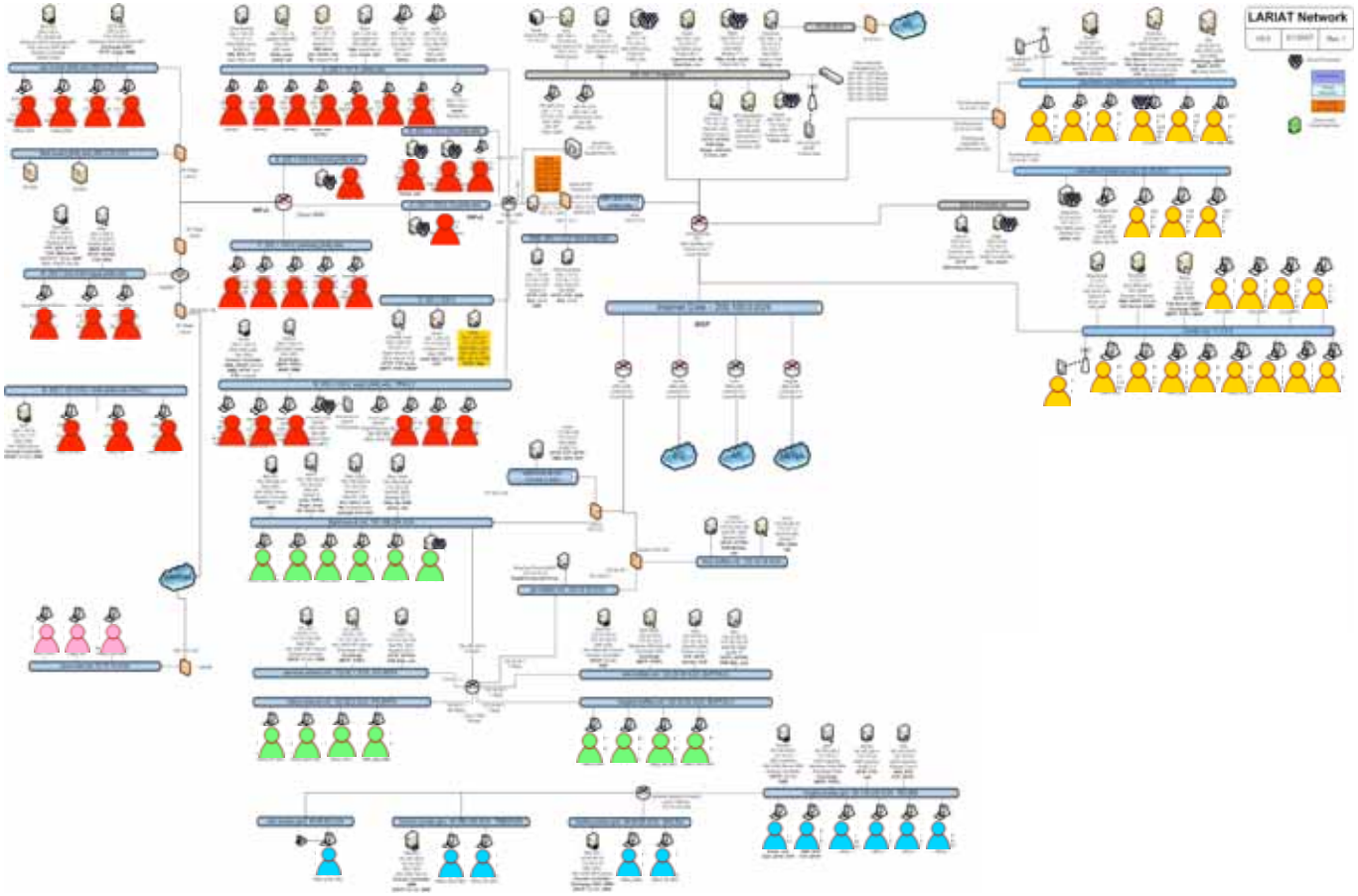
Integrated Environment Management for Information Operations Testbeds

Tamara Yu, Benjamin Fuller, John Bannick, Lee Rossey, Robert Cunningham

This work is sponsored by the Defense Advanced Research Project Agency (DARPA) and the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

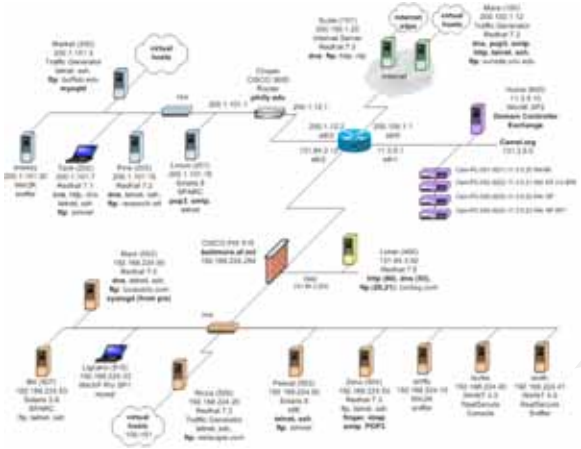
LARIAT Provides High-Fidelity Network Emulation and User Simulation

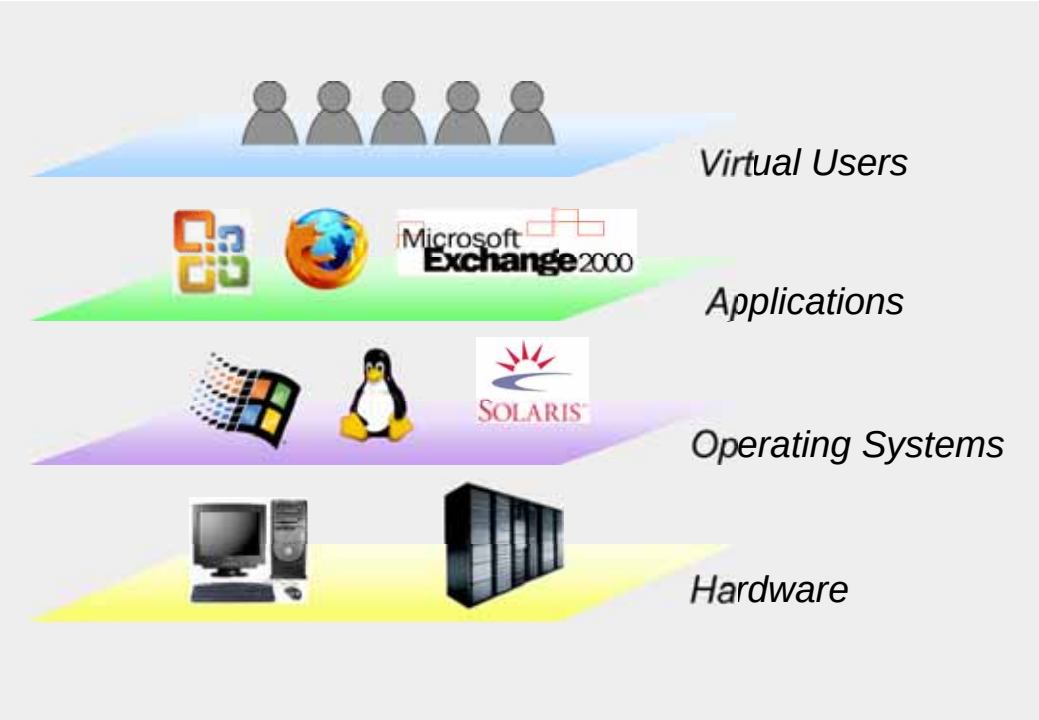


- LARIAT emulates all the users performing their daily duties
- Virtual users are overlaid onto existing physical networks
- Supports testbeds with 100s of nodes and 100,000s of users

Creating a Realistic Testbed is Challenging

- Testbed needs to be properly designed to meet the test objectives
- Configuration spans multiple levels: network, host, and user
- LARIAT works only if the underlying components work





A User Interface for Testbed Management

- **Two complementary approaches to effectively manage complex testbeds**
  - Intuitive user interface
  - Extensive automation

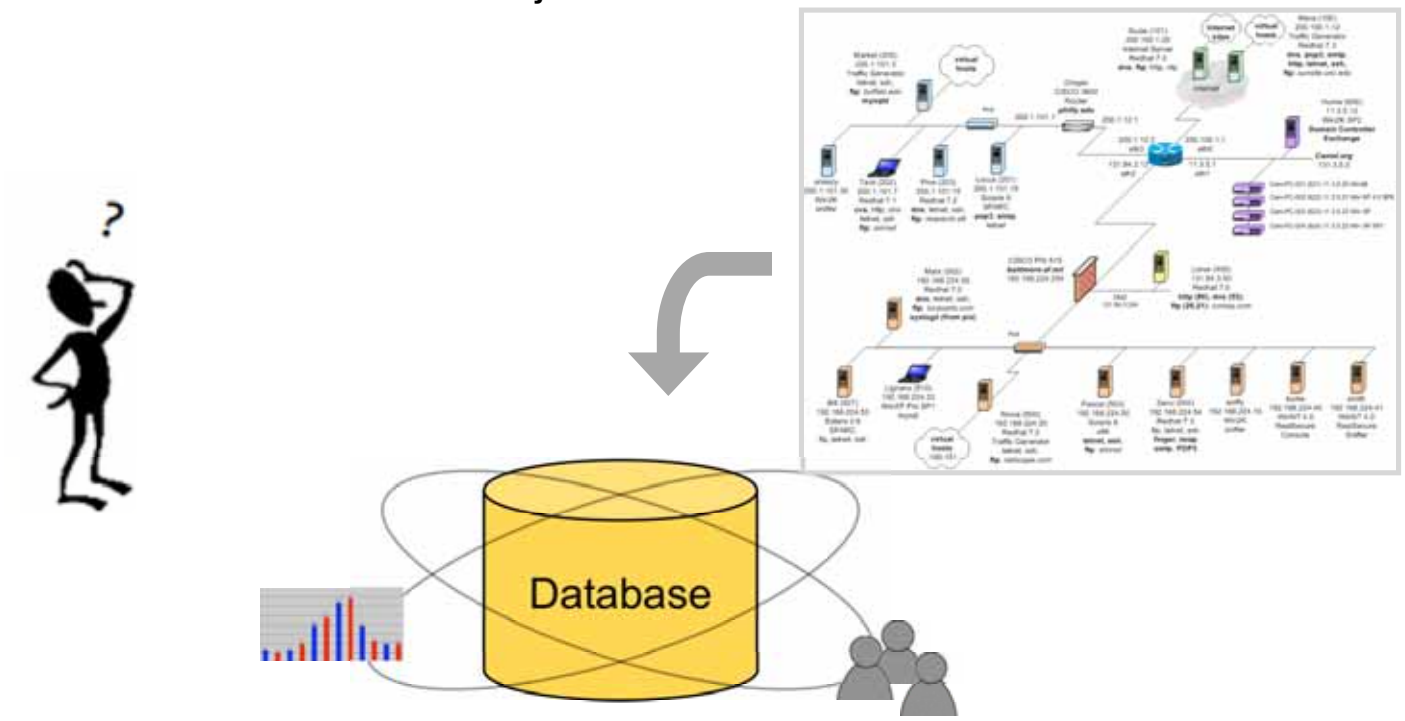
- **The Director simplifies the following key tasks of IO experiments**
  - Testbed specification
  - Testbed control (software deployment, troubleshooting and validation)
  - Testbed monitoring

Design Goals

- Appropriate System Abstraction
- Basic and Advanced Usage
- Scalable Performance and Visualization
- Interpretable and Correct Feedback

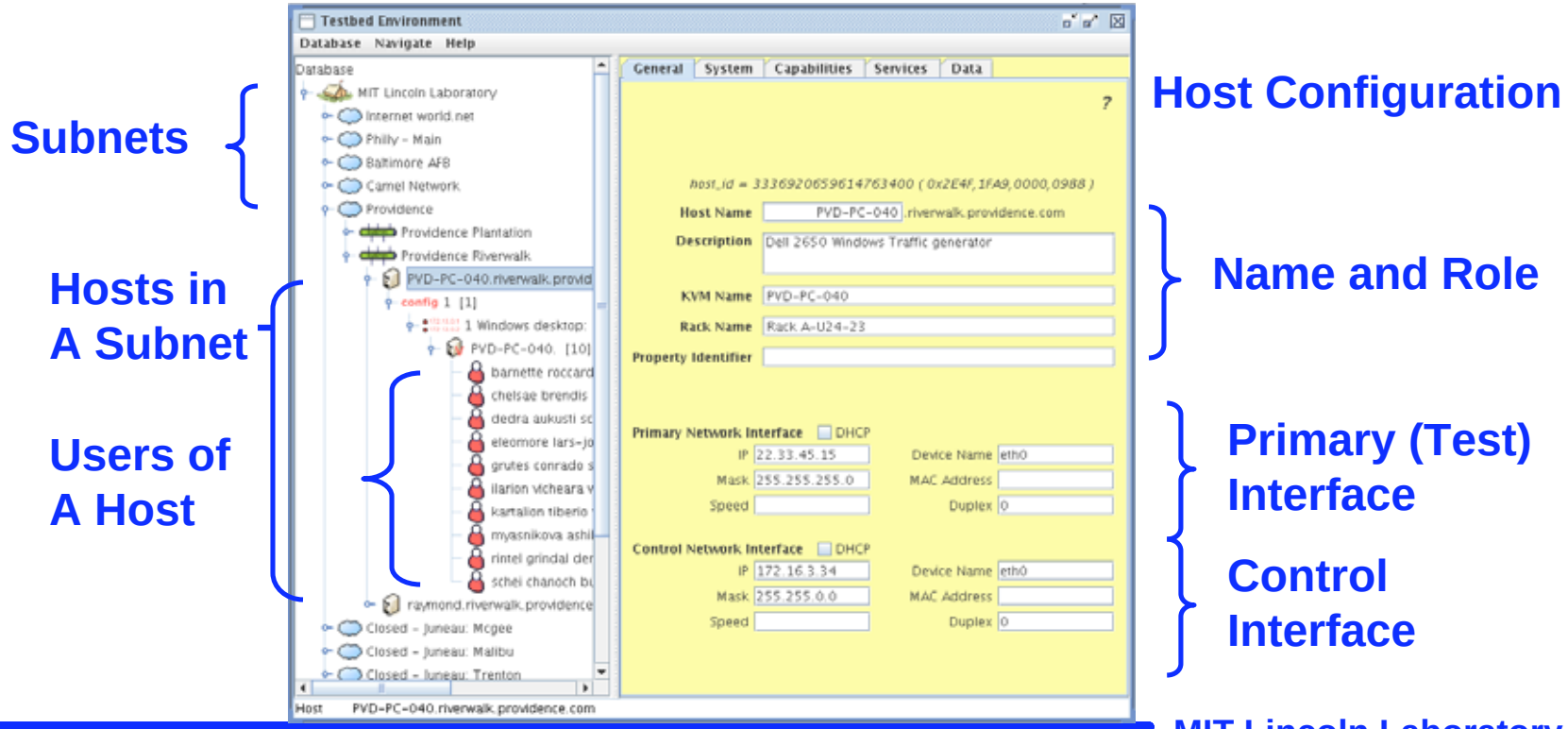
Test Definition Task

- **Configure LARIAT to emulate the desired test environment on the testbed network**
- **Challenges: complex configurations require a complex encoding**
  - Stored in a relational database with many interrelated objects and settings
  - Difficult to understand and adjust



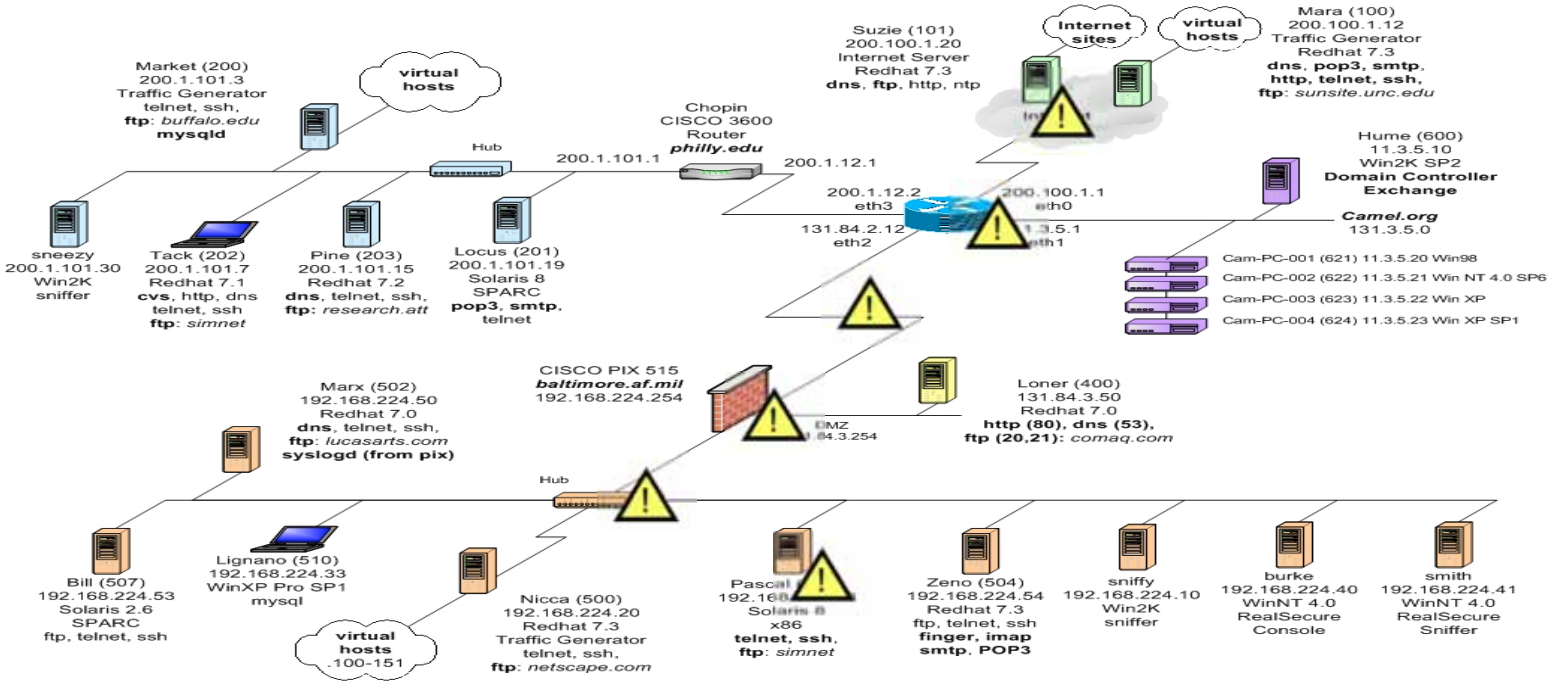
Test Definition Solutions

- **Automatically populate the database (U)**
  - Assign traffic profiles and server contents to hosts
  - Generate users and projects they share
- **Manipulate network objects in hierarchical context (A, S)**
  - Allows designers to easily view, modify, and allocate testbed resources



Diagnostics and Validation Task

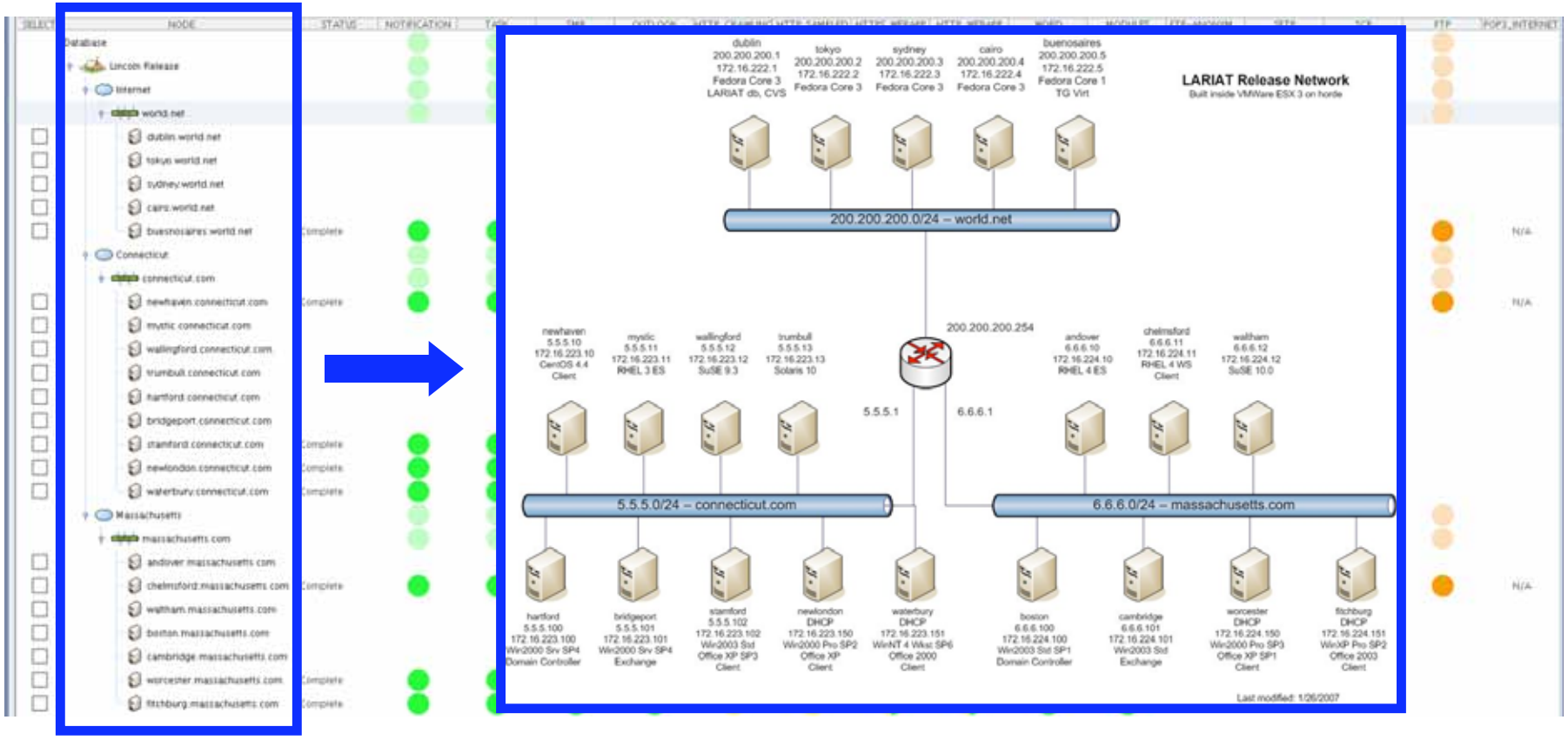
- **Validate that the testbed is set up correctly**
- **Diagnose and troubleshoot problems**
- **Challenges**
  - Large number of potential failure points (network, host, modules)
  - Difficult to pinpoint the cause due to network and software complexity



Diagnostic Solution: “Preflight”

(S)

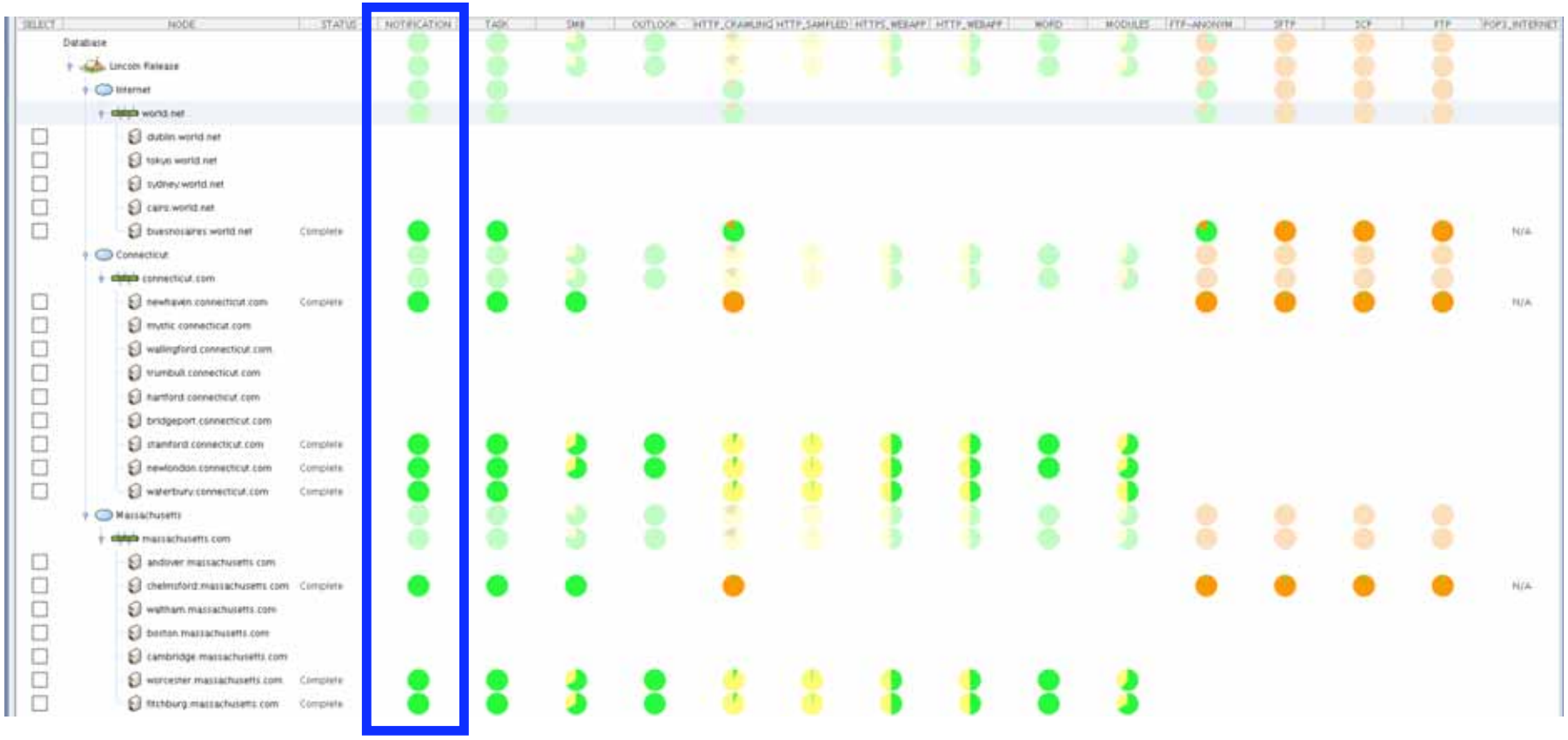
Network Nodes



Diagnostic Solution: “Preflight”

(F)

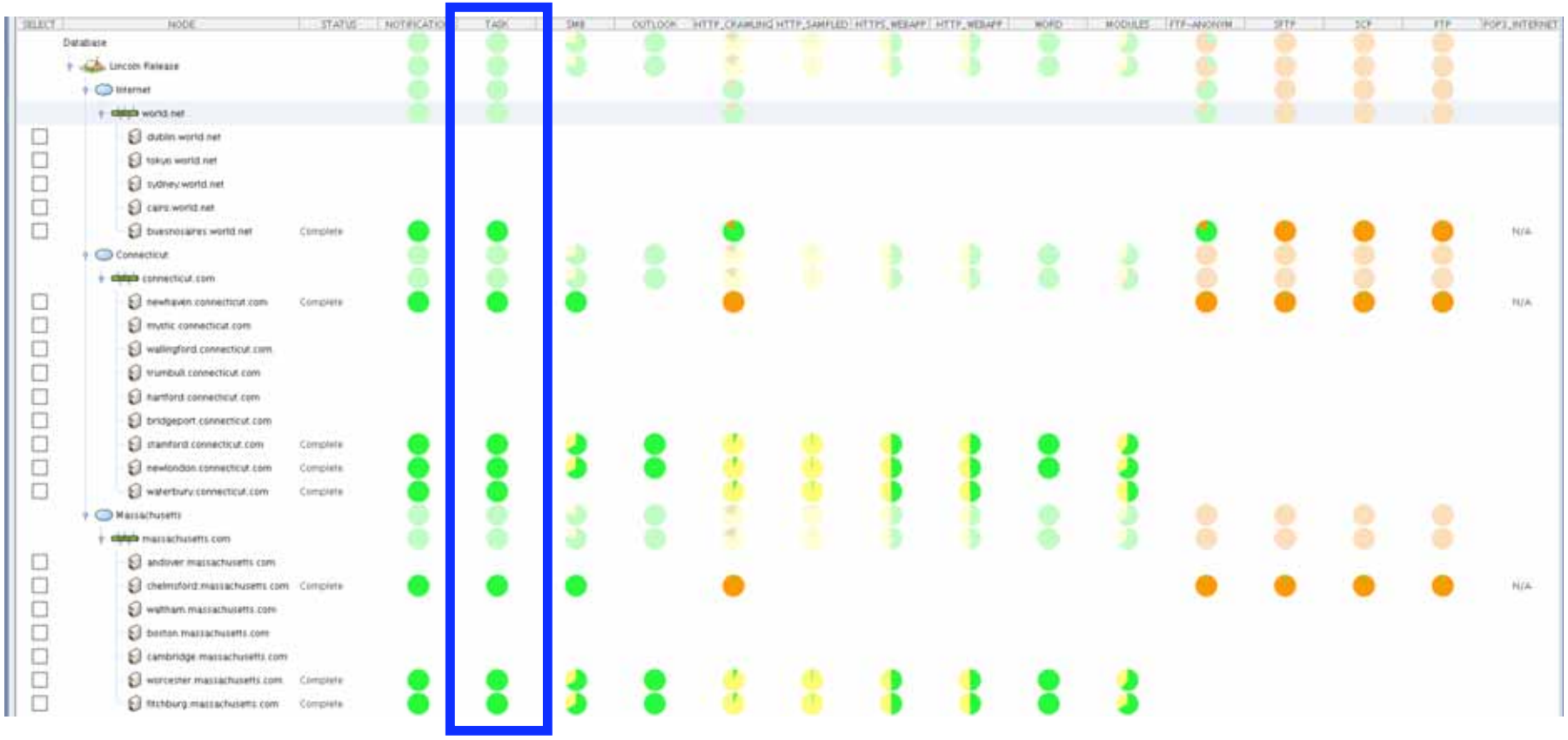
Task Notification Status



Diagnostic Solution: “Preflight”

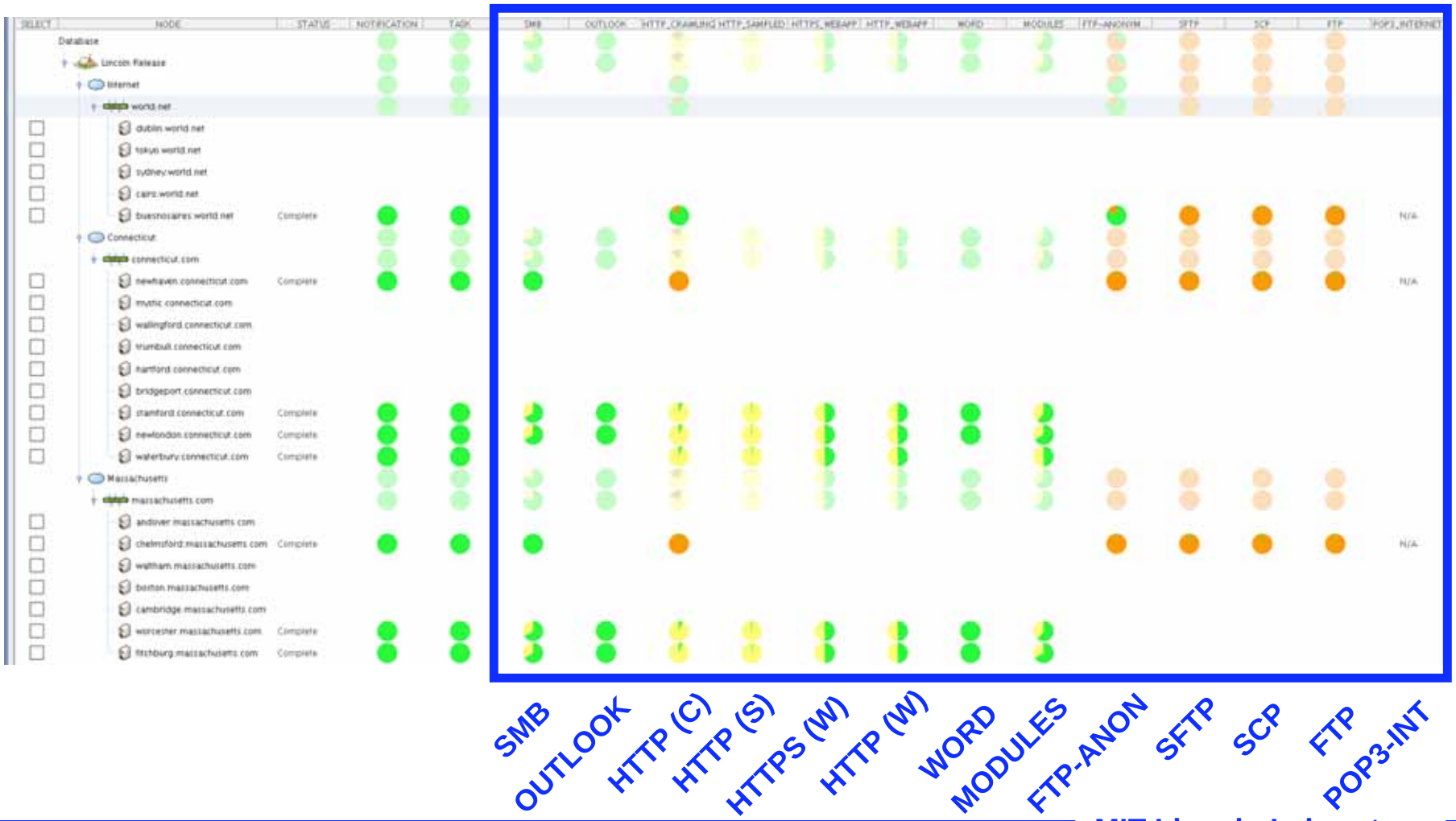
(F)

Task Execution Status



Diagnostic Solution: “Preflight”

Task Execution Status by Traffic Module (F)



Diagnostic Solution: “Preflight”

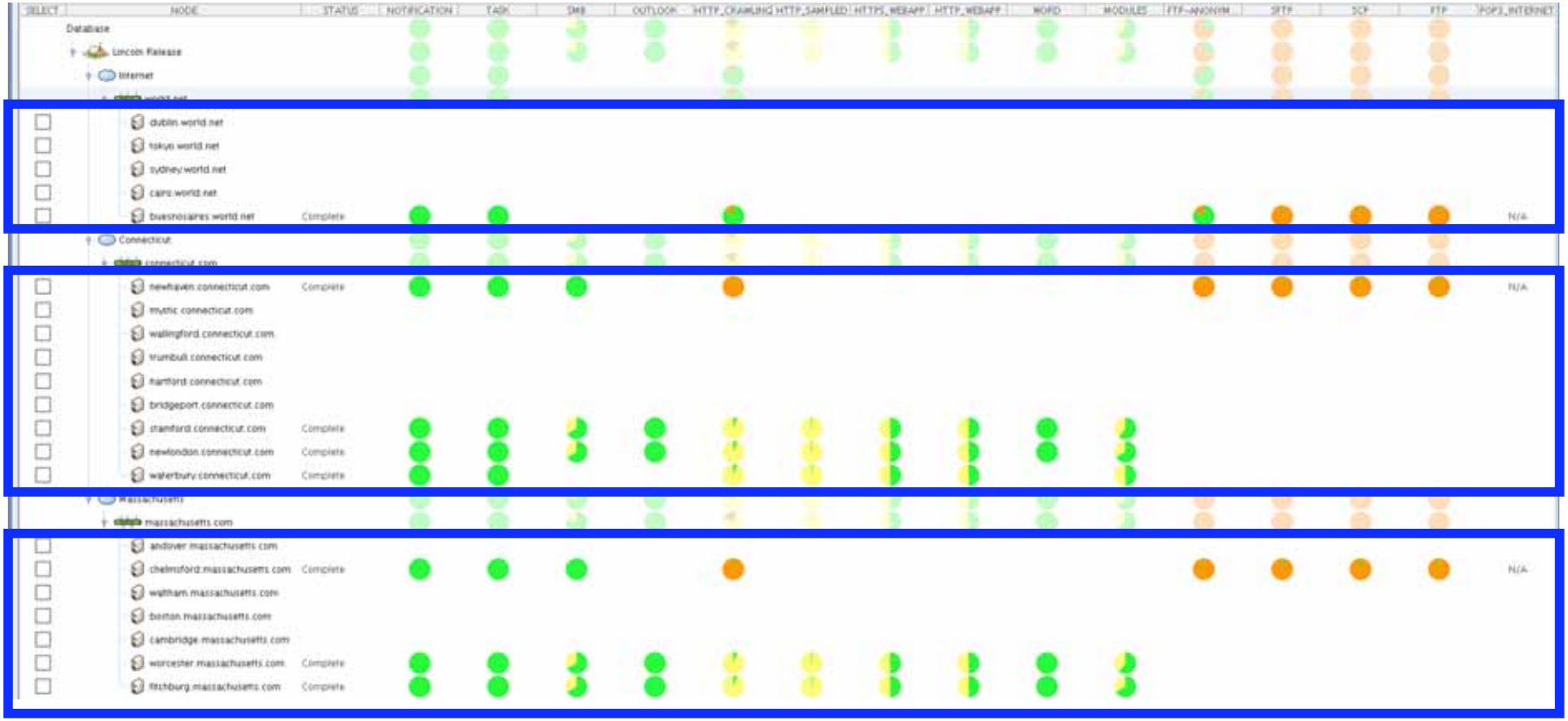
Mouse over status pie charts for popup details

(F)

Diagnostic Solution: “Preflight”

Task status by host

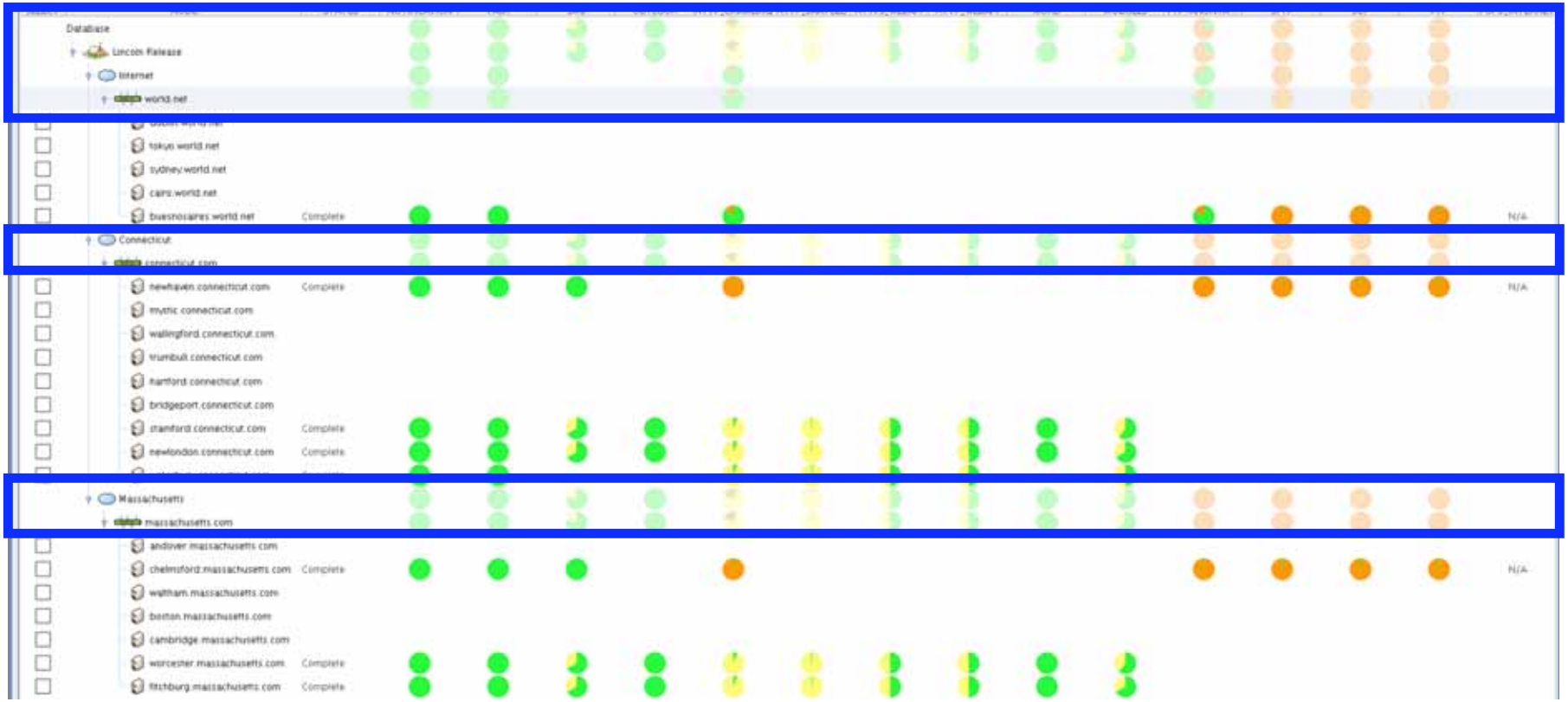
(F)



Diagnostic Solution: “Preflight”

Task status aggregated by subnet, network, and site

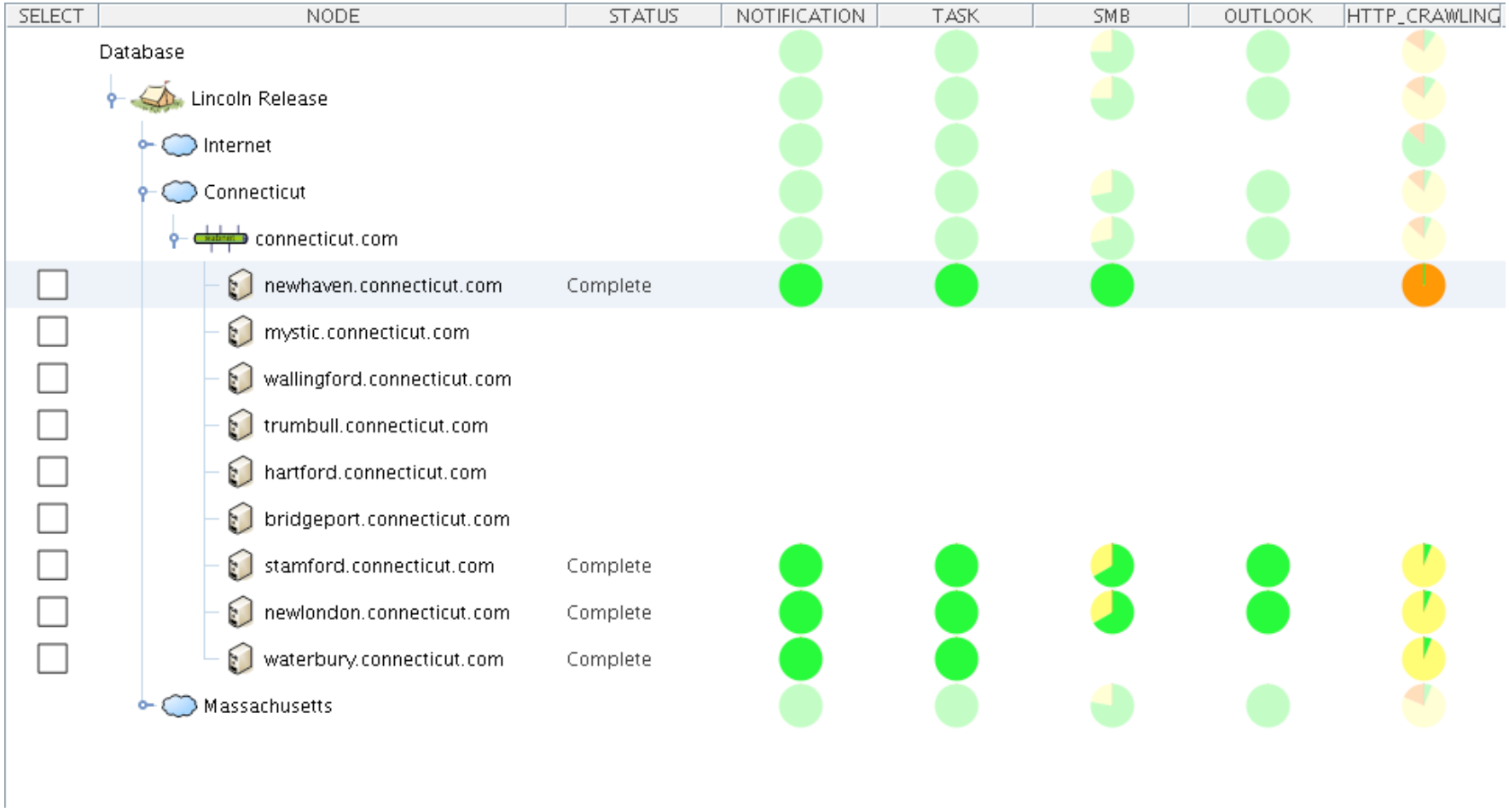
(S, F)



Diagnostic Solution: “Preflight”

Drill down to pinpoint errors (Particularly useful for large networks)

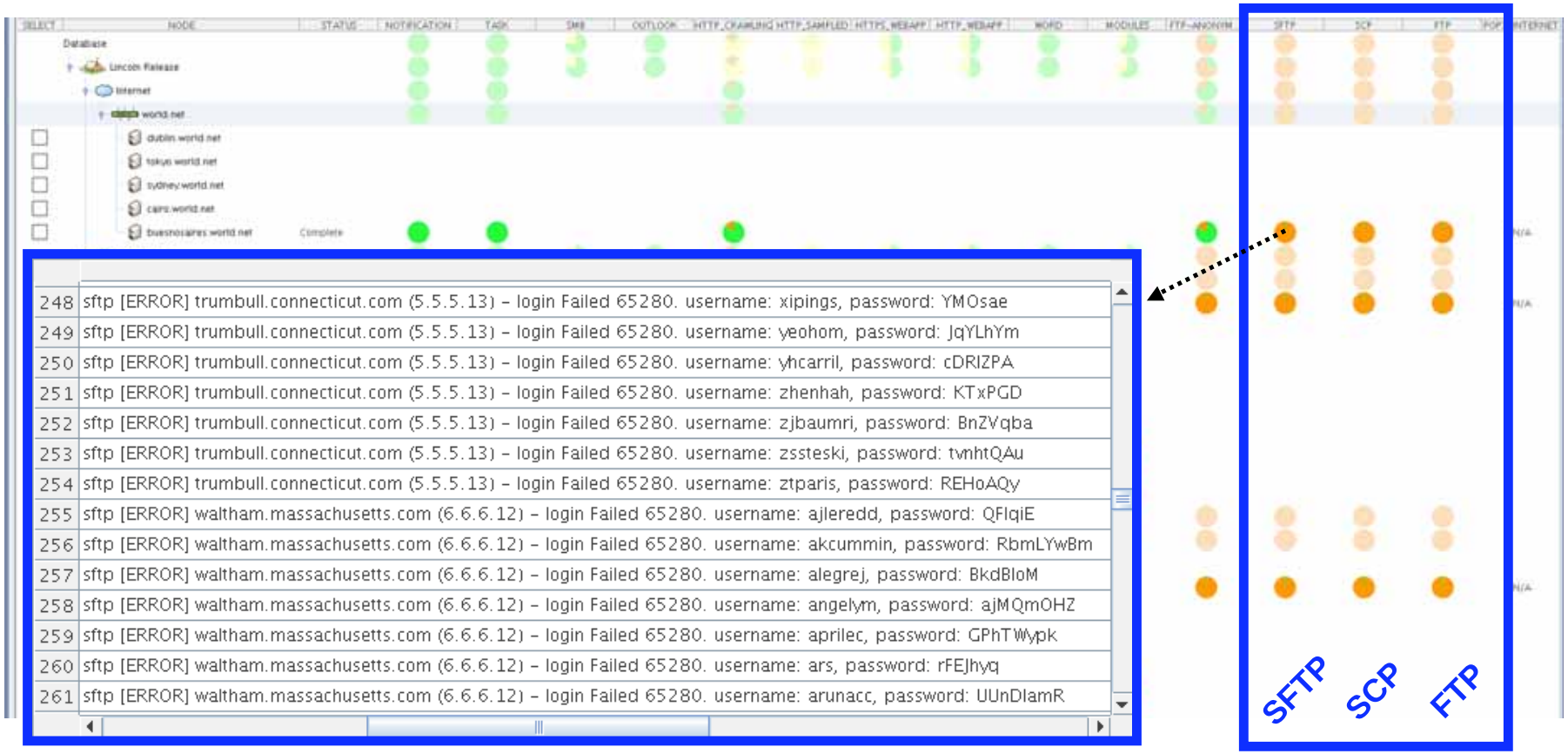
(S, F)



Diagnostic Solution: “Preflight”

User accounts were not loaded on the file transfer servers? **(F)**

Services were not started?



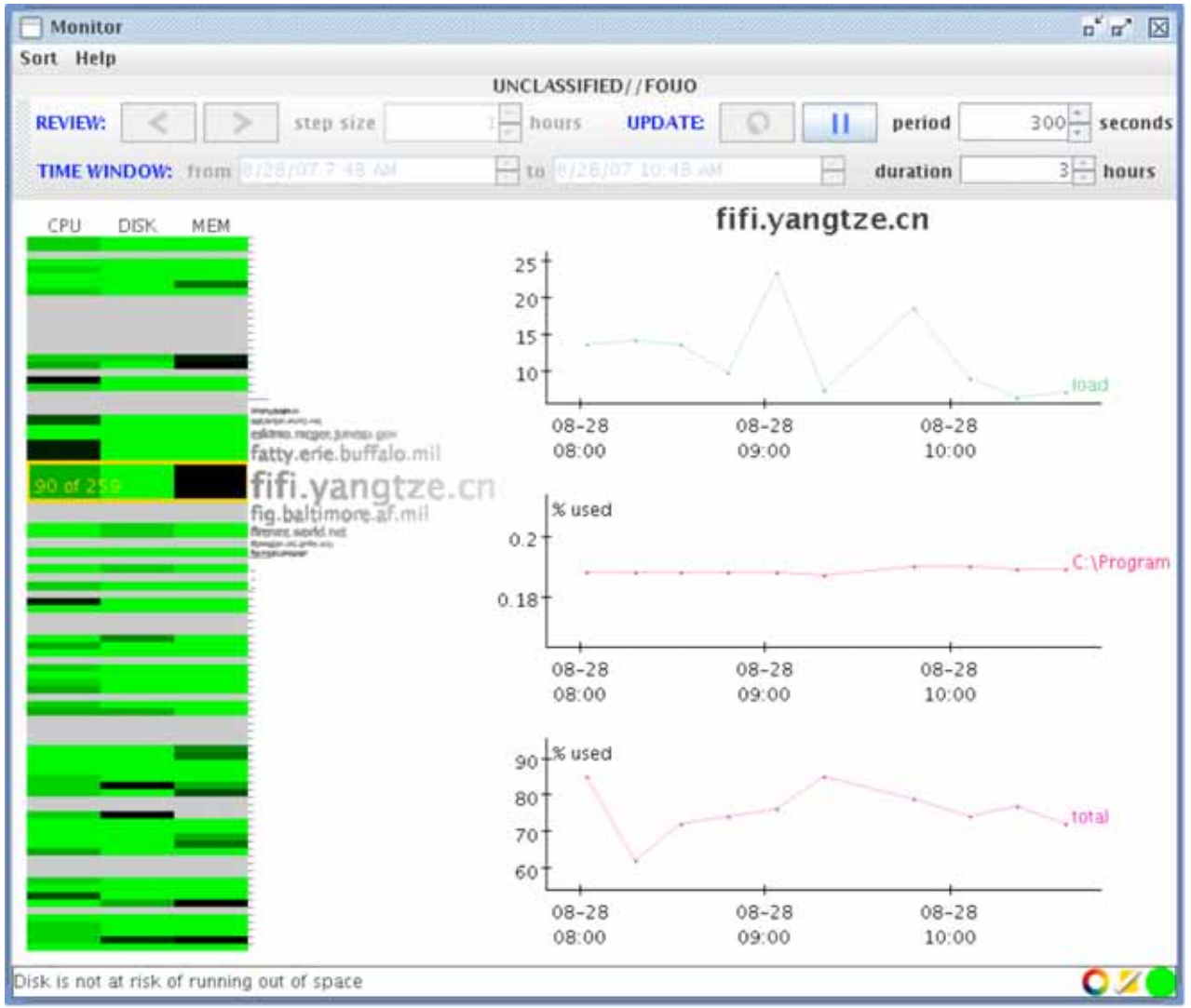
Test Monitoring Task

- **Continuously monitor testbed hosts to ensure normal operation**
- **Allow operators to detect and mitigate potential problems**
- **Challenges**
  - Large number of hosts and frequent events or measurements
  - Real-time processing of information
  - Variable time window of interest

Test Monitoring Solution: Host Health

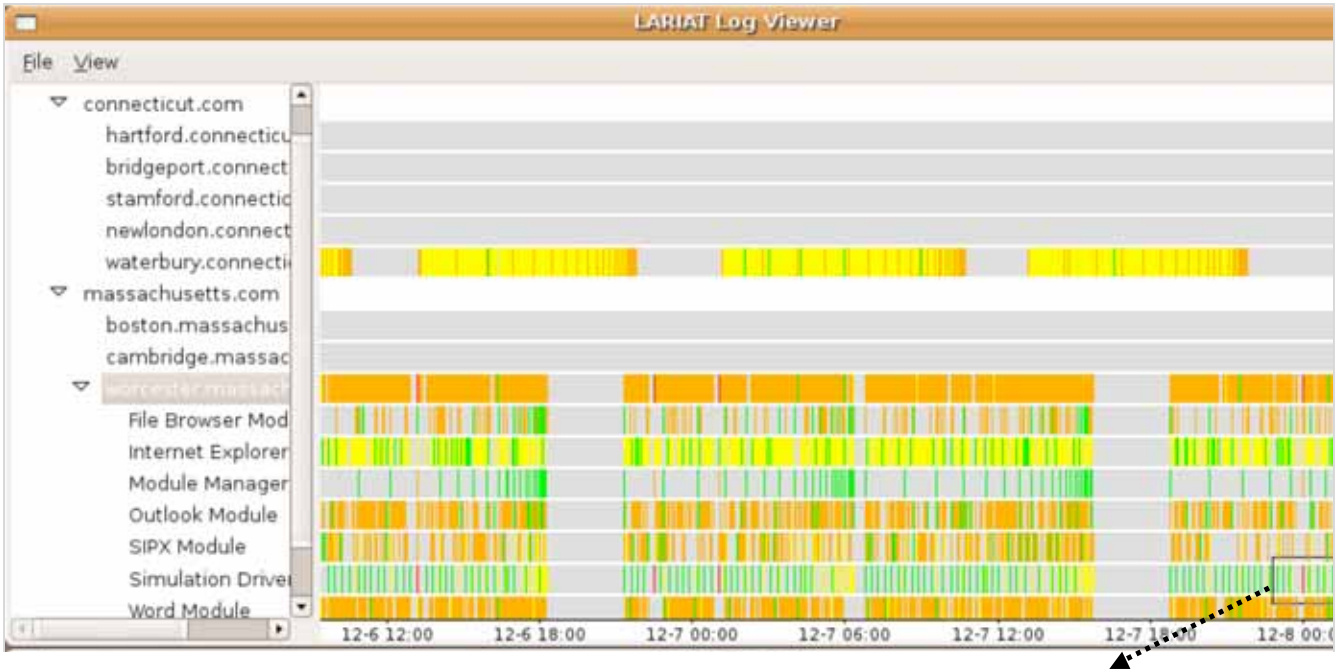
(S, F)

Testbed status at-a-glance



Usage history of selected host

Test Monitoring Solution: Traffic Status

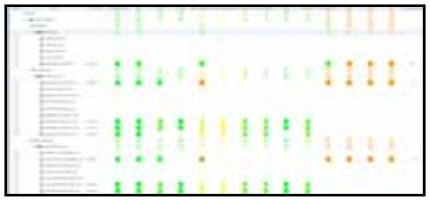
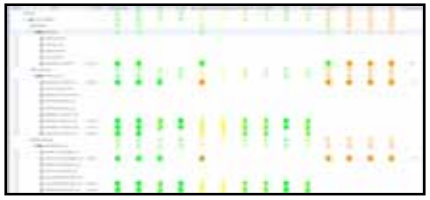


(S, F)

Host activity timeline

Timelines by module

Assessment

Appropriate System Abstraction	 	-Hierarchical model of the testbed environment -Unified host control mechanism masking underlying differences
Basic and Advanced Usage		-User parameters are auto-populated but can be fine-tuned -Basic parameters are generally shown first
Scalable Performance and Visualization	 	-Host health monitor can scale to arbitrary number of hosts without losing sight of trouble spots -Traffic health monitor can scale to arbitrary time window effectively
Interpretable and Correct Feedback		-Clearly displayed task execution status, categorized by network, hosts, subtasks, with counts and detailed log messages available on demand

Summary

- **LARIAT is a realistic, scalable traffic generation and testbed management tool for evaluating IO technologies**
- **The LARIAT Director, with the help of task automation**
  - Enables more complex and realistic test configurations
  - Significantly contributes to LARIAT's success
- **Today, it takes a few weeks to install, configure, validate, and baseline a large, complex testbed**
- **LARIAT is widely used for many types of IO tests**

Backup

Related Work

- **Experiment and Testbed Management Software**
  - Plush

    Framework for deploying distributed applications in PlanetLab
  - Experiment Workbench

    Framework for experiment management in Emulab
  - SEER

    User interface for the DETER testbed
- **Network Traffic Monitoring and Analysis Tools**
  - Network Eye

    Visualization of the inbound and outbound traffic of a "home" network
  - TNV

    Timeline visualization of network events by hosts and their corresponding connections and packet contents

Contributions

- **Developed an interface that enables experimenters to design and execute complex IO tests more easily using LARIAT**
  - Conceptual view of the testbed
  - Simplified operations
  - Centralized control and monitoring
- **Demonstrated the use of interface and visualization techniques in large-scale testbed management software**
  - Intuitive and scalable visual representations
    - *Hierarchical tree for testbed environment*
    - *Dynamic heatmap for host health*
  - Aggregation with visual cues for high-level assessment
    - *Task status view*
    - *Host health view*
  - Overview-to-details for troubleshooting