Predicting the Resource Consumption of Network Intrusion Detection Systems

Holger Dreger
Siemens AG
Corporate Technology

Anja Feldmann
Deutsche Telekom Labs /
TU Berlin

Vern Paxson
UC Berkeley /
ICSI

Robin Sommer
ICSI /
Lawrence Berkeley Lab

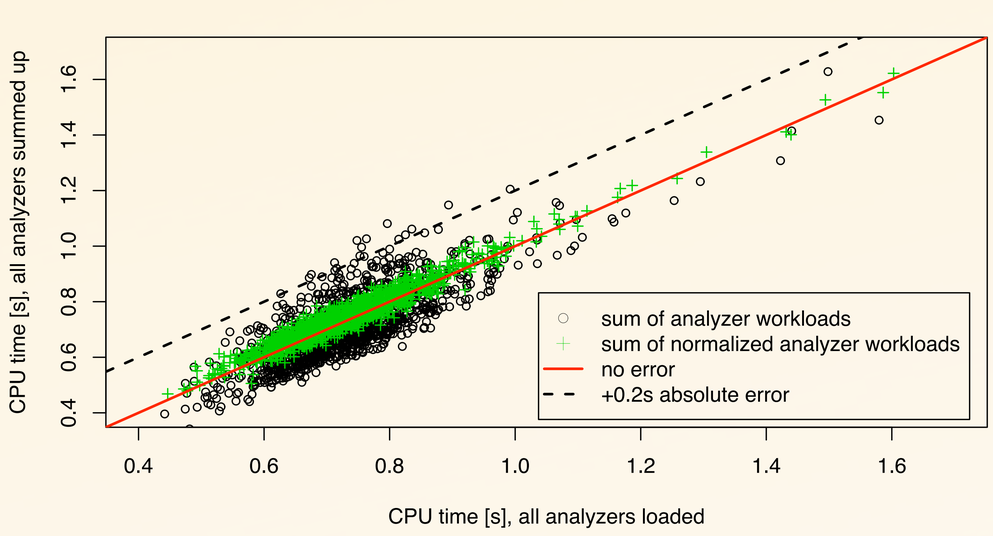
Motivation

- NIDS face trade-off:
  - Depth of analysis vs. resource usage.
- Many tuning options are hard to choose:
  - Relationship to resource usage unclear.
  - Variety in traffic requires headroom.
- Deployment becomes trial-and-error.
  - Often takes weeks to converge.
- **We devise an approach for resource prediction that provides operators with a sound starting point for NIDS deployment.**

Modeling NIDS Resource Usage

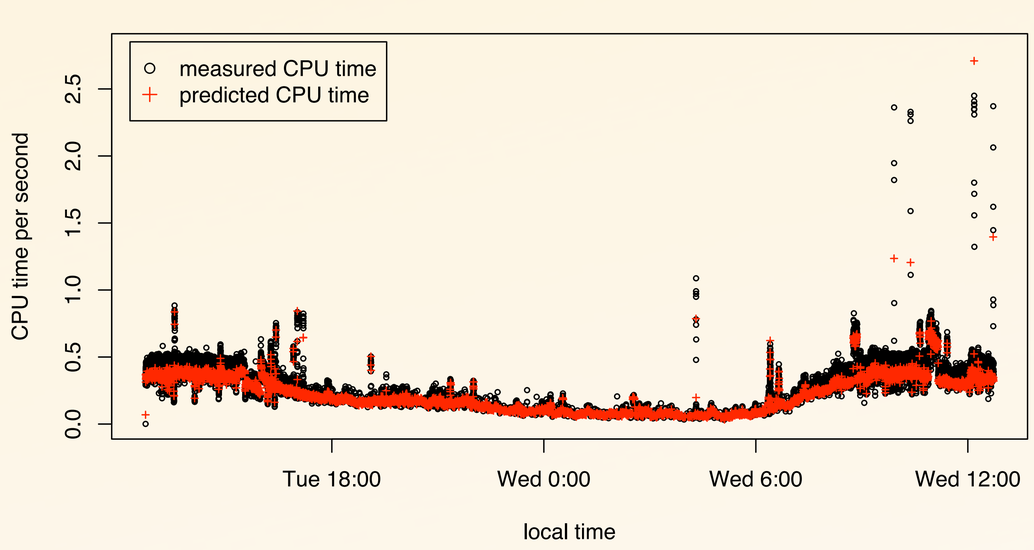
- A NIDS consists of many subcomponents for different classes of traffic (e.g., TCP, UDP, HTTP, SMTP, ...).
- We assume *orthogonal decomposition:*
  - Components use resources independently.
- Then we determine NIDS resource usage by
  1. Measuring average cost per class.
  2. Measuring traffic mix.
  3. Estimating total cost by scaling contributions.
- **Idealized model which we expect to capture resource usage well though not perfectly.**

Examining the Bro NIDS



- Validating our model with a complex NIDS.
- Using traffic from border of major campus:
  - 10Gbps link, 50,000 hosts, 2-4TB/day.
  - 24-hour trace (3.2TB, 137M connections).
- Verifying independence of resource usage:
  - CPU times sum up as expected (cf. plot).
  - Memory requirements do so mostly as well.
- Verifying that scaling is linear with number of connections:
  - Run on sampled trace & extrapolate.
  - Works well with simple configurations.
  - Slight overestimation with complex configurations.
- Challenges:
  - Sensitivity to measurement inaccuracies.
  - Rare activity hard to assess.
  - Differences between online & offline operation.
- **CPU and memory can generally be modeled well with a linear model (with a few caveats).**

Predicting Resource Usage



- Goal: Predict performance of configuration to
  - Expose trade-offs of different configurations.
  - Estimate when resources will get insufficient.
- Built tool to identify suitable Bro configurations.
- User provides:
  1. Short-term traffic sample (tens of minutes).
  2. Long-term connection-level log (days).
  3. Limits for CPU / memory usage (quantiles).
- Our tool `nidsconf` determines:
  1. Set of Bro configurations feasible with trace.
  2. *Long-term* resource prediction from connection log.
- Challenges:
  - Extrapolation of rare activity.
  - Prediction of user-defined analysis.
- **Comparing prediction with actual usage shows close match.** (cf. plot; basic configuration)

SIGMETRICS 2008
June 2-6, 2008, Annapolis, Maryland, USA