Making Sense of the Performance Riddle

Glenn O'Donnell

Principal Product Marketing Manager
EMC Software Group

odonnell_glenn@emc.com

Computer Measurement Group 2005

06-Dec-2005

A Riddle Wrapped in a Mystery Inside an Enigma

- Service performance is poorly understood, let alone managed, in the extreme majority of IT organizations
- Understanding this riddle requires navigating a complex labyrinth of interrelated technology and business elements
- New distributed services exacerbate this scenario
- Complexity is far beyond human comprehension, so automation technology, process, and standardization are the keys to simplifying and controlling service performance
- Enlightened organizations have proven success possible
- Difficult cultural shifts are needed to attain peak performance
- Automate, but beware of ambitious auto-adaptation (for now)

What is Performance?

a. Infrastructure

- Are the “nuts and bolts” of IT working “well enough”?

b. Applications

- Are the applications providing adequate results for our users?

c. Services

- Applications can mimic services, but what about workflow services?

d. Organizational

- Is the whole organization improving its own effectiveness?

e. Personal

- How am I helping to contribute to the goals of the organization?

f. Financial

- Do we contribute to revenue growth or expense reduction?

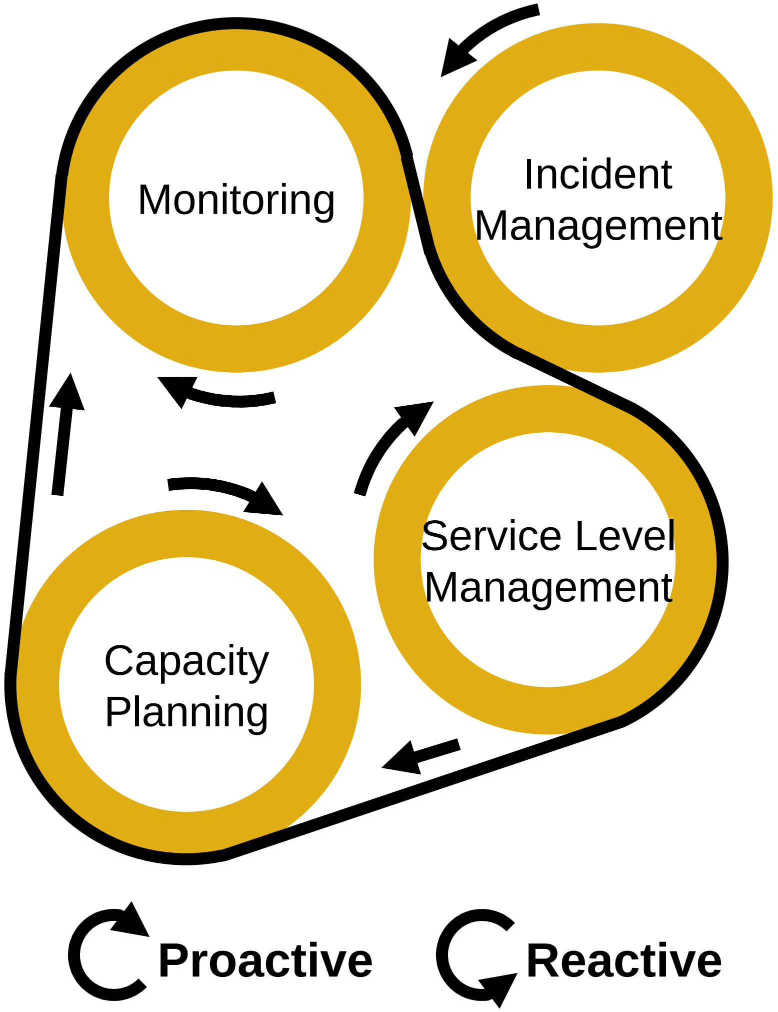
g. All of the above (plus more!)

Performance Must Be Quantifiable and Meaningful

- We need a **direct** means to measure the **right** metrics
- Generating reports can be meaningless if not in context
  - Action is the goal, not merely generating reports
- Data becomes information only if it is meaningful
- Focus on service metrics first and **then** infrastructure metrics
  - Infrastructure is only relevant in how it impacts services
  - Applications matter to end users; infrastructure does not
  - Don’t forget the performance of the organization itself!
- Unfortunately, 90 to 95% of collected data is useless
- Meaningful data is **related to** service performance

The Performance Management Process(es)

- Identify
  - Determine and model data domain constraints and relationships
- Detect
  - Detect anomalous conditions
- Collect
  - Collect for analysis and planning
- Process
  - Apply algorithms to analyze actual business impact of conditions
- Act
  - Reactive or proactive depending on the urgency of analysis conclusions



The Essential Role of Process

- Process initiatives such as ITIL highlight the importance of understanding and optimizing service performance
- All of the standard processes are relevant, but mostly:
  - Service Level Management
    - This is the fundamental real-time performance process
  - Capacity Management
    - Proactive action to preempt performance issues
  - Incident Management
    - Incidents are increasingly performance related, not hard failures
  - Problem Management
    - Chronic problems are also more commonly performance issues
- Processes offer discipline to maintain optimum performance
- Tools automate process execution and enforce discipline

Choose Your Analysis Appropriately

- Different methods serve different purposes

$$\bar{\bar{x}} \mp z_{1-\alpha/2}\frac{s_w}{\bar{n}\sqrt{m}}$$

- Performance data must be analyzed to provide context and meaning

$$C(p) = \frac{p}{1 + \sigma(p - 1) + \sigma\lambda p(p - 1)}$$

- Statistical baselining is becoming more commonplace
- Tying data to topological relationships helps provide relevance to the broader end-to-end service

$$F_n \equiv \sum_{k=0}^{N-1} f_k e^{-2\pi i n k/N}.$$

- Time-domain analysis can extract hidden patterns in data over time

$$f(\nu) = \mathcal{F}_t[f(t)](\nu) = \int_{-\infty}^{\infty} f(t)\, e^{-2\pi i \nu t}\, dt.$$

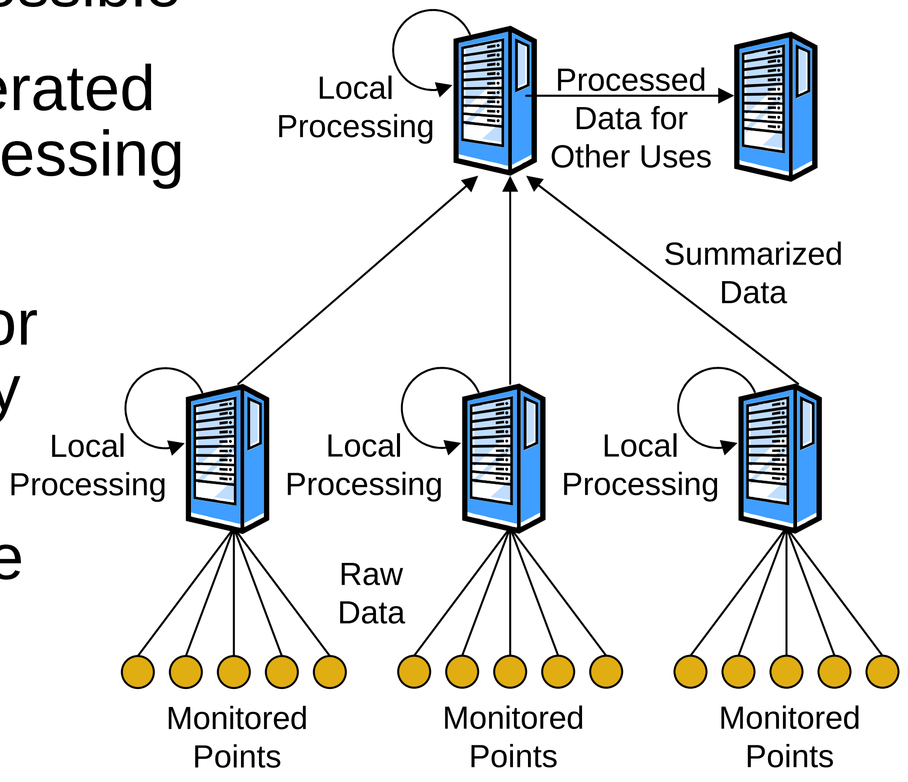
- Behavioral modeling will eventually put more data in context to its role in the overall system

$$\bar{\bar{x}}_l = \frac{1}{n-l}\sum_{j=l+1}^{n}\bar{x}_j$$

- Remember, >90% of all data is useless!

Processing the Data: Divide and Conquer

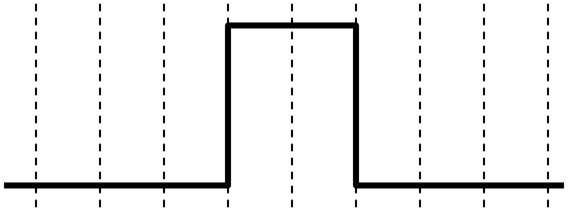
- Collecting raw data into a single processing point is wasteful
- Distribute processing where possible
- Architecture must support federated cooperation of distributed processing elements (even algorithms)
- Processed data is then used for incident management, capacity planning, and other purposes
- Ultimately, we want to drive the processing load all the way to the monitor point of interest (self-analysis)



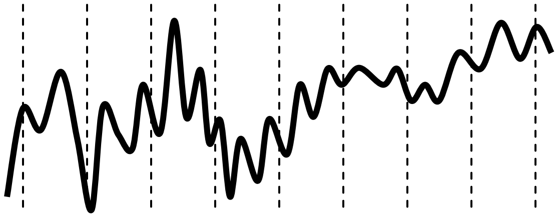
"Digital" Events vs "Analog" Performance

- Events are binary (condition has or has not occurred)
- Performance is variable

  … however …

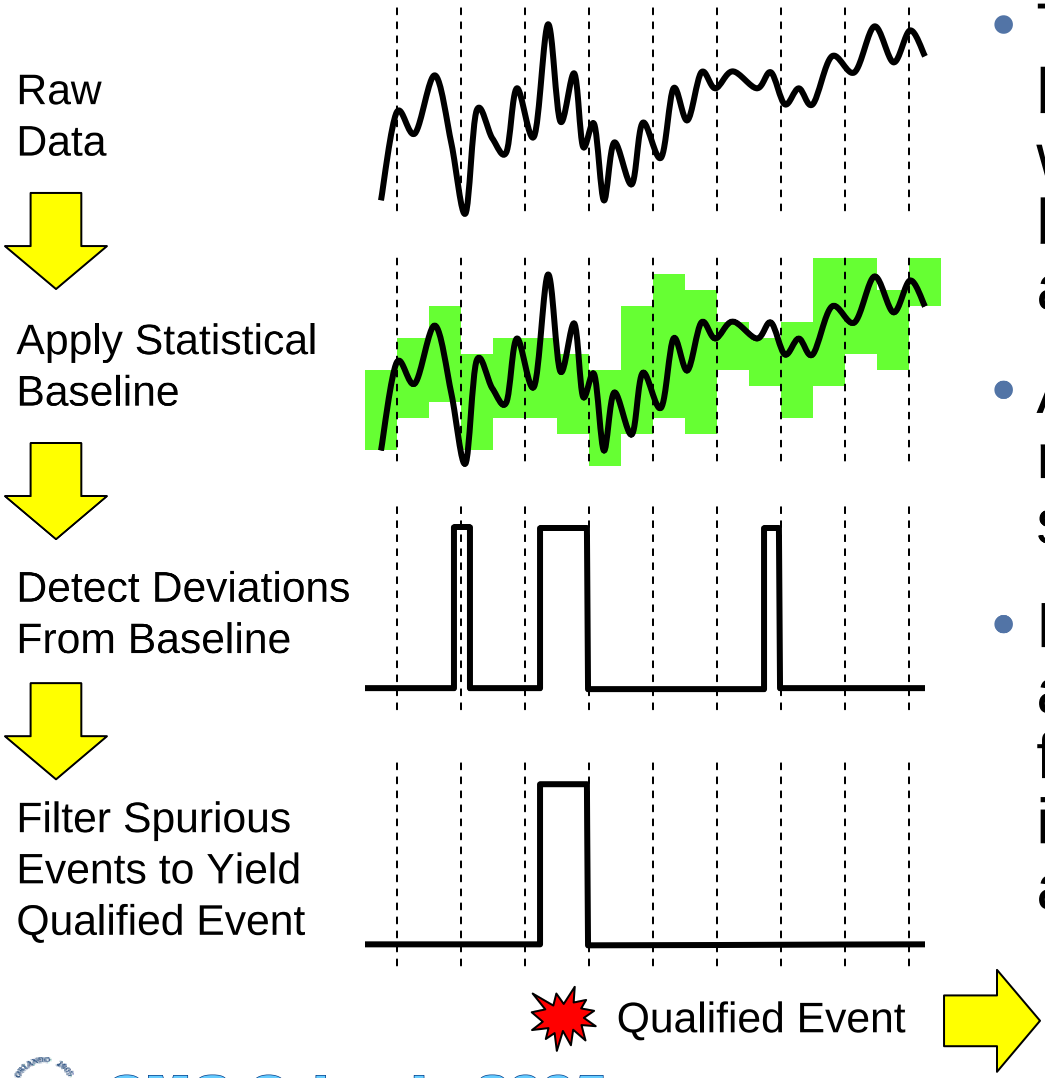


- Counting events yields an analog value
- Anomalous conditions of a varying quantity are binary events
- In truth, everything in the world is **<u>both</u>** analog **<u>and</u>** digital
  - Depends on time domain and context



- Event management tools and methods have become mature and pervasive
  - Use them for performance event processing too!

Use Event Management to Process Performance Data



Raw Data

Apply Statistical Baseline

Detect Deviations From Baseline

Filter Spurious Events to Yield Qualified Event

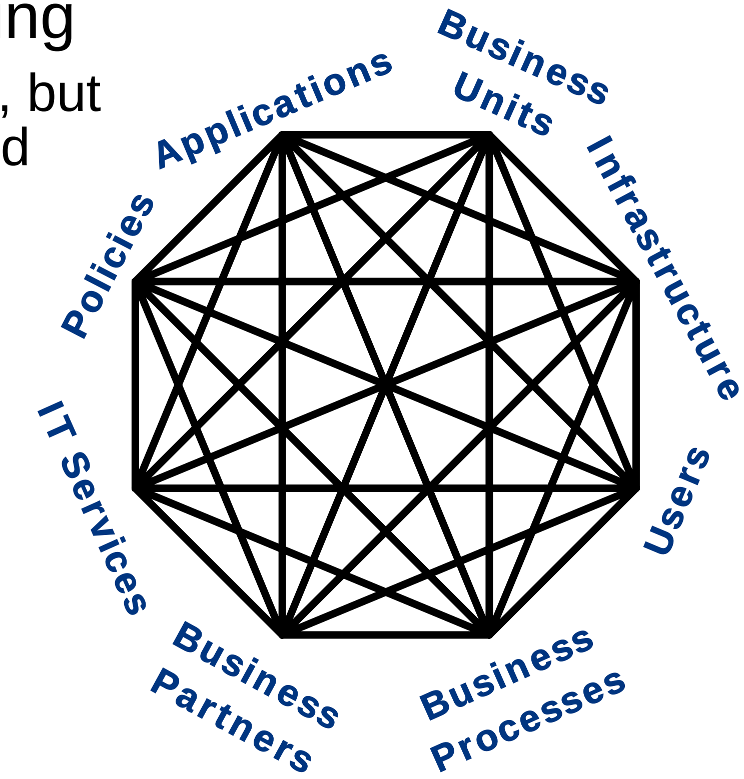
Qualified Event

Event Management and Correlation Tools

- The resulting event must be analyzed in context with other events that may be other performance anomalies or hard failures
- All of these events only matter if the business service is impacted
- Event correlation algorithms must account for relationships between infrastructure, applications, and business services

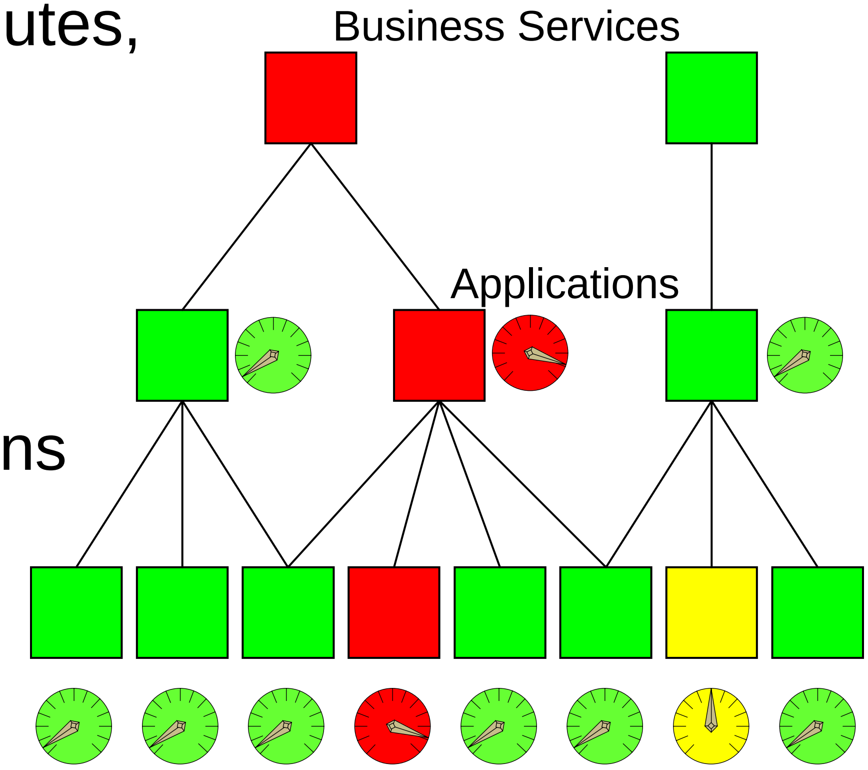
Relationships: The Intelligent Glue of Complex Systems

- Relationships give meaning to collections of objects
  - ANY collection in the universe, not just in IT!
- Manual mapping has long existed so the current emphasis is on automated discovery and mapping
  - Most discovery cannot be automated, but a lot can in infrastructure domains and applications are now the new focus
- Essential for multivariate analysis to analyze abstractions and assess business impact
- Proactive modeling exercises are impossible without relationships



Build a Model Based on Relationships

- Objects in the model have attributes, but so do relationships
  - Reflect actual dependencies and behavioral propagation
- A good model should represent the entire service chain from infrastructure through applications to business services
- This model becomes the core of configuration management
  - The embodiment of the “CMDB”



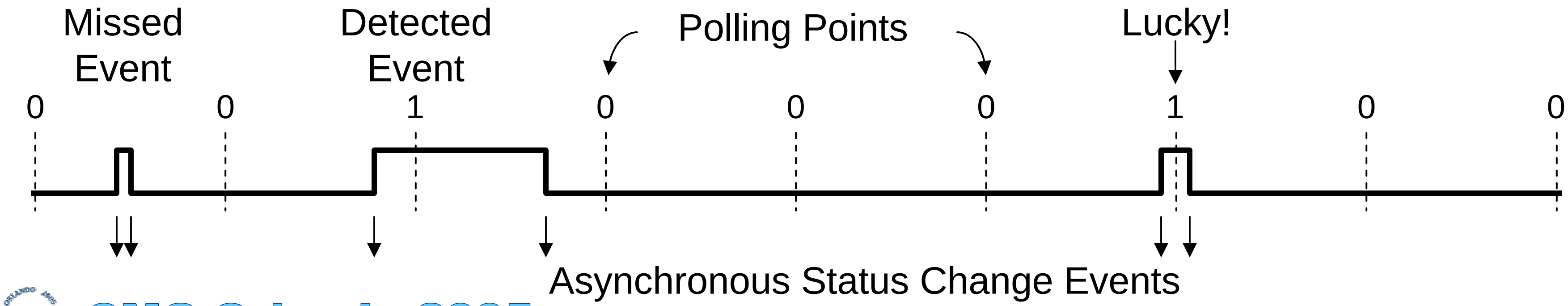
- As all IT functions fundamentally depend upon configuration, all automation functions depend upon an accurate model

The Perils of Polling

- Each component can produce hundreds of data points
  - Retrieving too much data wastes processing and storage resources
  - Tailor to poll only the minimum needed to provide value
- Representing the necessary relationships presents an n^2 problem with scalability where n = number of data points
- Data and anomalous events between samples are lost
  - Polled data is merely a snapshot or smoothed average
- Attaining “five nines” of availability is impossible with polling
  - 99.999% availability allows only 26 sec/mo of **cumulative** downtime
  - Nyquist rate: must poll every 13 secs to catch a single 26 sec event
- There **is** a better way! (although we cannot totally avoid polling)

Asynchronous Anomaly Notification (a.k.a., don't ask, please tell!)

- Generate real-time notifications for insight to actual behavior
- Fewer data points under normal conditions reduces processing and storage resources
  - Services under stress will increase data points, but processing algorithms can account for these pattern changes and summarize
- Requires some level of localized processing (thanks Gordon Moore!)
- SNMP Traps work, but CIM and WSDM are better choices
  - The industry needs to give genuine support to these standards



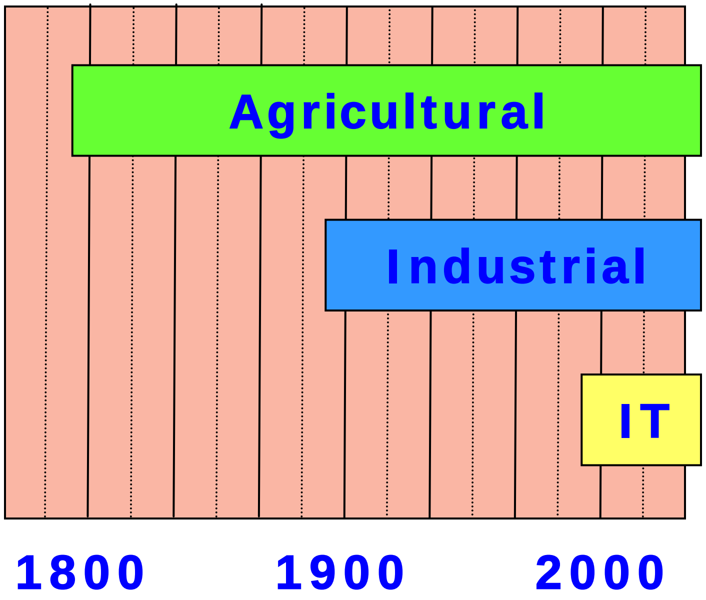
What To Do When Things Go Wrong

- Broad analysis of the IT service environment is necessary to answer the question of “Why?”
- Myopic focus on particular elements will be misleading
- Follow structured processes (e.g., ITIL Incident Management)
- Ask, “What changed?” … institute configuration and change management to drive ALL processes
- Learn from the experience!
  - Use a library of known problems for ongoing refinement
  - Long-term patterns will emerge, but active feedback accelerates this
- Attempting to answer this without automated assistance perpetuates destructive and chaotic response behaviors
  - Automate analysis, escalation, notification and process workflow

Laziness is the Mother of Invention!

- Automate every possible task
- Reduces costs and errors
- Automation has been central to IT since our beginning
  - IT is, simply put, business automation
- We are moving now to the next phase of automation
  - Configuration discovery
  - Change execution and Provisioning

Automation Timeline (history repeats)



- Auto-adaptive IT remains science fiction, but …
  - By 2010, many of today's skills will be obsolete

How to Ensure Good Performance in the Future

- Only consider individual components in the context of their impact upon higher-level service abstractions
- Engineer performance via model-based capacity planning
  - Optimize performance of services, not individual components
- Institute structured processes to instill discipline
  - Adopt ITIL as a starting point, but avoid ITIL religion
- Automate execution with the **right** tools to enforce discipline
- Negotiate service levels with business users and develop a catalog of services along with SLAs for each
  - Track performance of each service
  - Don’t forget to include organizational performance as SLOs
  - Communicate compliance to users regularly and adapt as needed

A Few Suggestions for Future Development

- We must abandon polling as a data collection mechanism
  - Does not scale well to large environments
  - Fails to achieve high-reliability monitoring
  - Will take time, so polling will remain a part of our near-term future
- Localize processing (hyper-distributed management) and generate asynchronous notifications upon anomaly detection
- Agree to standards and promote their realistic adoption
  - Object models and web services messaging amongst objects
- Expand research into multivariate & time-domain algorithms
  - Will further automate broader end-to-end service analysis
- **Change the way you think about performance!**
  - **It’s not always technical & rarely about individual components**

EMC^2

where information lives

For a copy of this presentation,

please contact:

odonnell_glenn@emc.com