Simplifying Performance Data

Visualizing application data in real time

Graphical Data Visualization

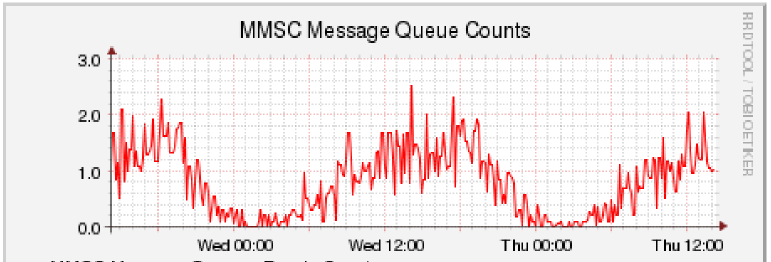
- Graphical data can provide insight into the overall performance and service level for an application in a way that is accessible to many different types of users

We make mobile work.

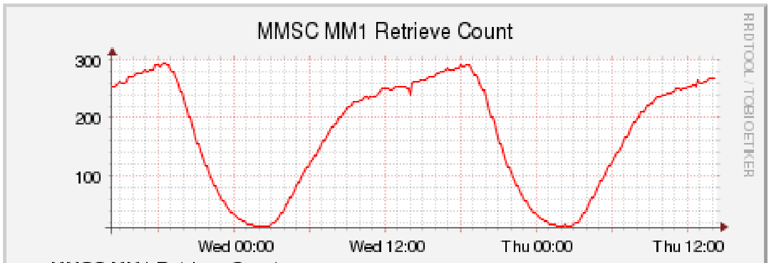
2 day | Dashboard - Default | Browse | Status | Logout

NMS - Default Dashboard

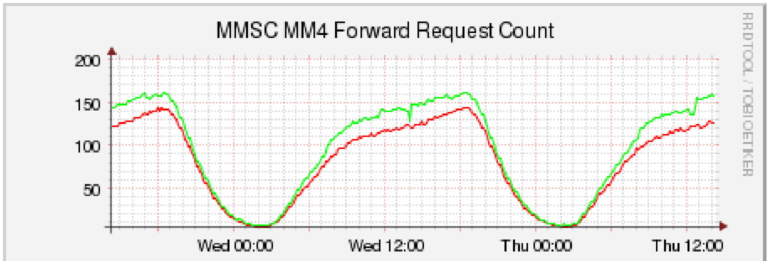
Graph End Time: 2012/05/03 14:13:00, Graph Frame: e-15min, Baseline for same time frame over 5 weeks						
Graph	**Data Source**	**Current Rate**	**Baseline Rate**	**Rate Variance**	**Last Update**	**Export**
MMSC MM1 Retrieve Count	MMSC MM1 Retrieve Count (ops/sec)	265.00	252.74	+4.9%	0 seconds	csv xls
MMSC MM4 Forward Request Count	MMSC MM4 Forward Request Recieved Count (ops/sec)	155.80	133.88	+16.4%	0 seconds	csv xls
MMSC MM4 Forward Request Count	MMSC MM4 Forward Request Sent Count (ops/sec)	123.70	122.32	+1.1%	0 seconds	csv xls
MMSC Message Queue Counts	MMSC Message Queue - Ready Count (messages)	1.15	1.14	+0.7%	0 seconds	csv xls



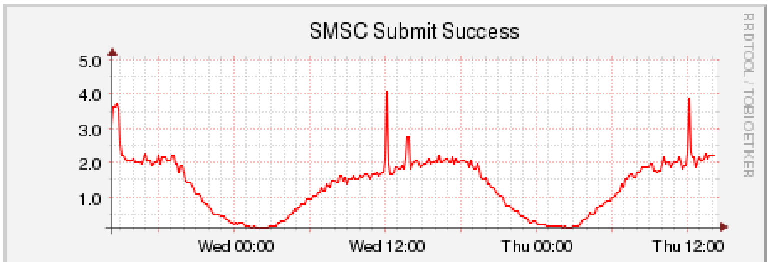
🟥 MMSC Message Queue - Ready Count
Max: 2.533 Avg: 795.859 m (messages)
Frame: 2 day(s)
End: Thu May 3 2012 14:13:00 PDT



🟥 MMSC MM1 Retrieve Count
Max: 293.907 Avg: 170.019 (ops/sec)
Frame: 2 day(s)
End: Thu May 3 2012 14:13:00 PDT



🟥 MMSC MM4 Forward Request Sent Count
Max: 144.151 Avg: 81.111 (ops/sec)
🟩 MMSC MM4 Forward Request Recieved Count
Max: 160.914 Avg: 94.180 (ops/sec)



🟥 SMSC Submit Success
Max: 4.050 Avg: 1.308 (ops/sec)
Frame: 2 day(s)
End: Thu May 3 2012 14:13:00 PDT

We make
mobile work.

Application Data

- Application KPIs – simple numbers
  - 100 Messages in a file queue
  - 400 Connections to the database
- Application KPIs – incrementing counters
  - 300 packets per second
  - 20 messages per second
  - 1.5 seconds average to submit a message

Transaction Rates

Sendmail application log example

```
Dec 8 05:40:50 localhost sendmail[9443]: oB85ZsiT000318: to=<+15557654321/TYPE=PLMN@example.com>, delay=00:00:01, xdelay=00:00:01, mailer=smtp, pri=12071, relay=[10.11.12.13]#mx1.example.com [10.11.12.13], dsn=2.0.0, stat=Sent (2.0.0 Ok: queued as 1352DC00658AE)
```

Multiple counters can be tracked by email domain and delivery status.

Average Transaction Time

Sendmail application log example (continued)

```
Dec 8 05:40:50 localhost sendmail[9443]: oB85ZsiT000318:
    to=<+15557654321/TYPE=PLMN@example.com>,
    delay=00:00:01, xdelay=00:00:01, mailer=smtp,
    pri=12071, relay=[10.11.12.13]#mx1.example.com
    [10.11.12.13], dsn=2.0.0, stat=Sent (2.0.0 Ok: queued
    as 1352DC00658AE)
```

Using two counters an average time can be calculated

We make mobile work.

2 day ▸

Dashboard | Browse - Graph View | Status | Logout

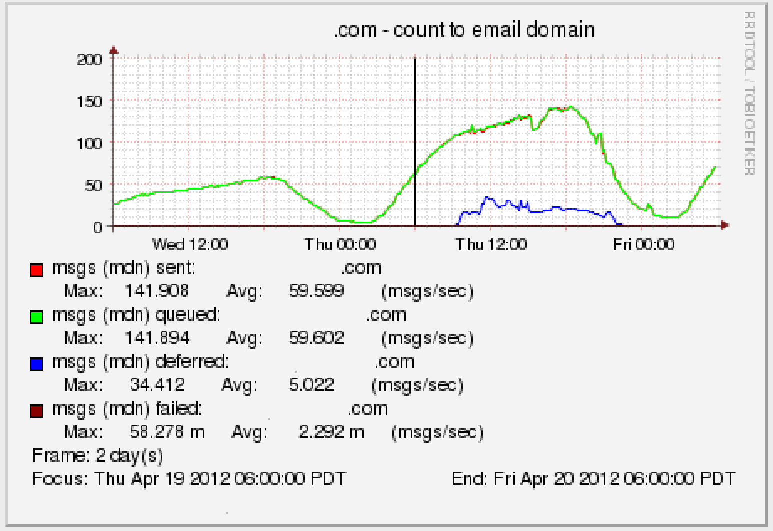
Clear Node Group Filter

- icmms-dnt-mqueue
- icmms-eswitch
- icmms-lc-mqueue
- icmms-loghost
- icmms-mm7-adapter
- icmms-mmsc
- **icmms-mqueue**
- icmms-nms
- icmms-web

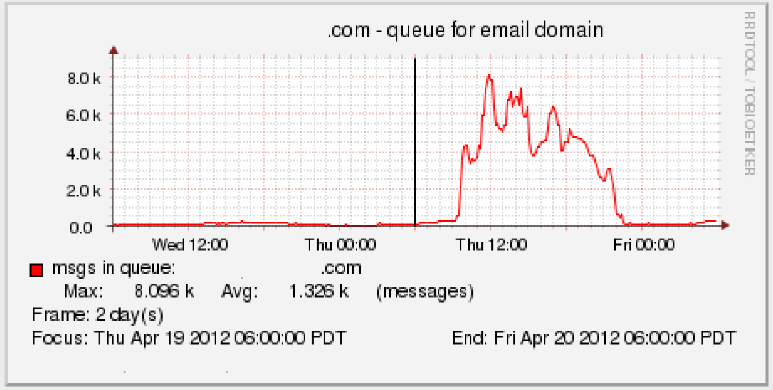
Add Node Filter

All
icmms-mqueue001
icmms-mqueue002
icmms-mqueue003
icmms-mqueue004
icmms-mqueue005
icmms-mqueue006
icmms-mqueue007
icmms-mqueue008
icmms-mqueue009
icmms-mqueue010
icmms-mqueue011
icmms-mqueue012
icmms-mqueue013
icmms-mqueue014
icmms-mqueue015
icmms-mqueue016
icmms-mqueue017
icmms-mqueue018
icmms-mqueue019
icmms-mqueue020
icmms-mqueue021
icmms-mqueue022
icmms-mqueue023
icmms-mqueue024
icmms-mqueue025
icmms-mqueue026
icmms-mqueue027

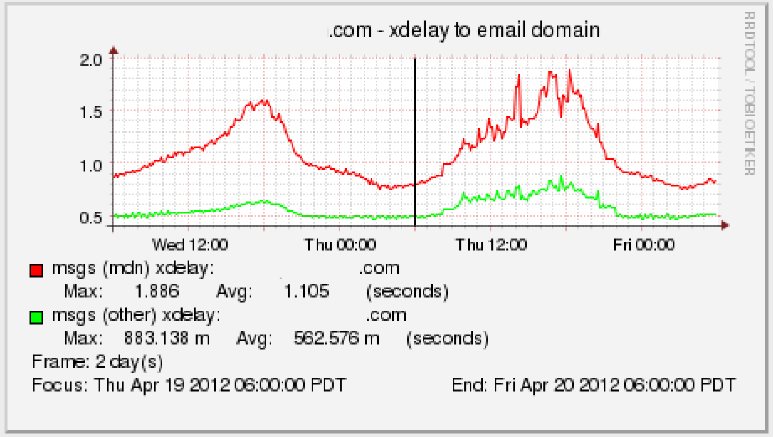
Add Page Rule Filter Any | Clear Graph Name Filter | (queue|mdncount|xdelay).*mm4. |conn.*



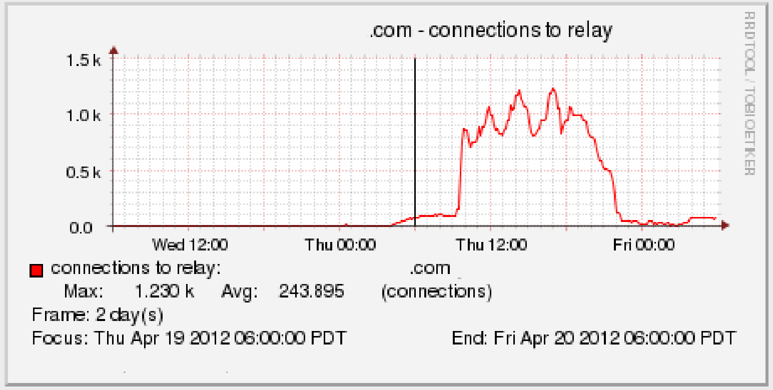
```
🟥 msgs (mdn) sent:                .com
     Max:   141.908     Avg:    59.599     (msgs/sec)
🟩 msgs (mdn) queued:              .com
     Max:   141.894     Avg:    59.602     (msgs/sec)
🟦 msgs (mdn) deferred:            .com
     Max:    34.412     Avg:     5.022     (msgs/sec)
🟥 msgs (mdn) failed:              .com
     Max:    58.278 m   Avg:     2.292 m   (msgs/sec)
Frame: 2 day(s)
Focus: Thu Apr 19 2012 06:00:00 PDT          End: Fri Apr 20 2012 06:00:00 PDT
```



```
🟥 msgs in queue:               .com
     Max:   8.096 k     Avg:   1.326 k    (messages)
Frame: 2 day(s)
Focus: Thu Apr 19 2012 06:00:00 PDT          End: Fri Apr 20 2012 06:00:00 PDT
```



```
🟥 msgs (mdn) xdelay:              .com
     Max:     1.886     Avg:     1.105     (seconds)
🟩 msgs (other) xdelay:            .com
     Max:   883.138 m   Avg:   562.576 m   (seconds)
Frame: 2 day(s)
Focus: Thu Apr 19 2012 06:00:00 PDT          End: Fri Apr 20 2012 06:00:00 PDT
```



```
🟥 connections to relay:               .com
     Max:   1.230 k     Avg:   243.895    (connections)
Frame: 2 day(s)
Focus: Thu Apr 19 2012 06:00:00 PDT          End: Fri Apr 20 2012 06:00:00 PDT
```

We make mobile work.

Data collection

- Pick a simple API to collect data if possible
  - CSV output file
  - JSON output file
- Use an integration layer to retrieve data
  - Allows for adapters to send to various graphing and alarming applications (Cricket, Cacti, Netcool, etc)

Data correlation

We make mobile work.

- Using graphs to visualize the effect of events in real time
  - Track the effect of events on application performance in real time
  - Correlate the behavior of multiple KPIs when compared together

Constant improvement

- Incremental steps
  - Break out monitoring enhancements into incremental tasks
  - Continually review data and events to find opportunities for improvement

Closing

We make mobile work.

- Adding graphical visualization can greatly simplify the complexity of monitoring applications
- Graph data provides a window that many different members of the organization can use to view application performance and find correlations