Monitoring @ Scale
in Salesforce

Kamil Smuga, Mihai Bojin

@kamilsmuga
@mihaibojin
Forward-Looking Statements

Statement under the Private Securities Litigation Reform Act of 1995:

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.
Kamil
Software Development Manager
Customer Experience Tools
Salesforce

“We help engineers and their customers observe and operate their services quickly and painlessly.”

- User Experience
- Product Management
- Front and Back End Developers

@kamilmuga
kamil@salesforce.com
What is monitoring?

According to Google…

“to observe and check the progress or quality of (something) over a period of time.”

Source: Google, March 20, 2017: https://www.google.com/search?q=what+is+monitoring&oq=what+is+monitoring
Why would you monitor software?

- Detection
- Proactivity
- Knowledge
A Standard Approach To Monitoring

- **collectd**
- **StatsD**
- **Nagios®**
- **ZABBIX**
- **sensu**
- **BASH**

Industry-standard Tools
Metrics Collection

Graphs

Alerts
How does it scale?
1-10 Servers

Life is good!
Hundreds of Servers
Life is OK
Thousands of Servers? Hundreds of Thousands of Servers?

Things get a bit more complicated…
Challenges

Scale and Visibility

Scale

● We require tools that scale
● Active feature development
● High performance

Visibility

● Millions of metrics don’t fit on one screen
● System metrics not relevant to customer experience
The Solution
“Customer-first” monitoring

- look at what customers experience
- dedicated (per-customer) metrics
- ability to monitor customer specific metrics
The Solution
Improved, open-source tools

- **Argus** = scalable, API first, **time-series data store**
- **Refocus** = visualizing system/service **health** and status
- **Pyplyn** (“pipeline”) = convert Argus time-series into Refocus health metrics
Scale @ Salesforce
## The reality of scale

### Growth Across The Clouds

<table>
<thead>
<tr>
<th>2014</th>
<th>2016</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>99.97%</strong> Availability</td>
<td><strong>99.98%</strong> Availability</td>
</tr>
<tr>
<td><strong>50</strong> Production Instances</td>
<td><strong>109</strong> Production Instances 118% ↑</td>
</tr>
<tr>
<td><strong>490 Billion</strong> Transactions</td>
<td><strong>1.1 Trillion</strong> Transactions 124% ↑</td>
</tr>
<tr>
<td>• 230ms average latency</td>
<td>• 210ms average latency</td>
</tr>
<tr>
<td><strong>10</strong> Data Centers</td>
<td><strong>20</strong> Data centers 100% ↑</td>
</tr>
<tr>
<td>• 1st in EMEA – London</td>
<td>• 3 in EMEA – London, Paris, Frankfurt</td>
</tr>
<tr>
<td><strong>194</strong> MC Customer Databases</td>
<td><strong>395</strong> MC Customers Databases 104% ↑</td>
</tr>
<tr>
<td><strong>247 Billion</strong> emails sent</td>
<td><strong>478 Billion</strong> emails sent 94% ↑</td>
</tr>
</tbody>
</table>
Fact

2014 to 2016

~100% growth

- production instances
- transactions
- data centers
Scale @ Salesforce

How do we run our service?

- Multitenancy
  - customers share infrastructure

- Instances
  - 5-8,000 orgs / instance
  - 100+ services running

EU5
Scale @ Salesforce

Transactions = Customer requests

- a few thousand metrics / transaction
- 5 billion transactions per day
- trillions of data points
How is a customer doing today?

Let’s call them “Hooli”
How is Hooli doing today?

Not a trivial answer
How is Hooli doing today?
Every incident/case is different

- Getting past the initial identification operations

- Performance metrics
  - Application metrics
  - Database metrics
  - System metrics
Application metrics

- Average Page Time
- CPU time
- Database Total Time
- Memcached
Application metrics

- Average Page Time
- CPU time
- Database Total Time
- Memcached
Application metrics

- Average Page Time
- CPU time
- Database Total Time
- Memcached
Application metrics

- Average Page Time
- CPU time
- Database Total Time
- Memcached
Database metrics

- DB CPU time
- DB active sessions
- Physical reads/writes
- and many others:
  - Buffer gets
  - Cluster Wait Time
  - Concurrency Wait Time
  - etc.
Database metrics

- DB CPU time
- **DB active sessions**
- Physical reads/writes
- and many others:
  - Buffer gets
  - Cluster Wait Time
  - Concurrency Wait Time
  - etc.
Database metrics

- DB CPU time
- DB active sessions
- **Physical reads/writes**
- and **many** others:
  - Buffer gets
  - Cluster Wait Time
  - Concurrency Wait Time
  - etc.
System metrics

- Uptime/last restart time
- Storage (SAN)
- Networking
- Load
- IOPS
- etc.
System metrics

- Uptime/last restart time
- Storage (SAN)
- Networking
- Load
- IOPS
- etc.

Graph showing SAN lag in minutes from March 13, 11:38 to March 14, 11:10 (GMT).
How is Hooli doing today?
Not a trivial answer

- Nothing wrong at first glance
- More logs
- What about other customers?
- Difficult to investigate using traditional monitoring techniques
The Solution
Customer-centric approach

Step 1: Collect relevant metrics

- Look at problems from the customers’ “eyes”
- Scale our tools to support the required volume of data: Argus (open-source)
- Collect metrics relevant to individual customer orgs
Customer-centric approach
Better, but not good enough!
Customer-centric approach

Step 2: Understand customer health

- What do these graphs mean?
- Obstacle: Interpreting the data
- Per-customer health dashboards: Refocus (open-source)

<table>
<thead>
<tr>
<th>INSTANCE</th>
<th>Hooli</th>
<th>Planet Express</th>
</tr>
</thead>
<tbody>
<tr>
<td>APT</td>
<td>955.4</td>
<td>218.8</td>
</tr>
<tr>
<td>CACHE</td>
<td>8.1</td>
<td>8.2</td>
</tr>
<tr>
<td>IO</td>
<td>4.7K</td>
<td>0.0</td>
</tr>
<tr>
<td>MEM</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NET</td>
<td>43.6K</td>
<td>15.7K</td>
</tr>
<tr>
<td>LOAD</td>
<td>162K</td>
<td>16.7K</td>
</tr>
<tr>
<td>LAG</td>
<td>3.1K</td>
<td>7.3</td>
</tr>
<tr>
<td>ERROR</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>LIMIT</td>
<td>5.4</td>
<td>0.0</td>
</tr>
<tr>
<td>LOGIN</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PERF</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>578</td>
<td>175</td>
</tr>
</tbody>
</table>
Customer-centric approach
One page view for many metrics
Conclusion
If There’s One Takeaway...
Key points

- Monitor what the customers are experiencing
- Use tools that can scale
- Get visual correlation
  - Common interpretation of results
  - High-level status
Demo

The setup
Software & Hardware stack

- **Raspberry Pi**, SenseHAT, DHCP server
  - Collect metrics (Python)
- **Argus** (time-series data store) (JAVA)
- **Pyplyn** (ETL, interprets the data) (JAVA)
- **Refocus** (visualization) (Javascript)
Demos

#1 SenseHat

- Collect *temp* + *gyro*
- Store in Argus, display in Refocus, ship with Pyplyn

#2 Open-source projects activity

- Collect repository activity
- Visualize
What is Argus?

- Time Series Data & Events
- Inbuilt Service Protection
- Flexible Dashboarding
- High Throughput
- Low Latency
- Horizontally Scalable
Argus REST API

- API-first design
- Decoupled UI
- Authenticated
What is Pyplyn?
Convert Argus metrics to Refocus visualizations
Pyplyn configurations

JSON-based ETL

```json
{
    "extract" : [ {} ],
    "transform" : [ {} ],
    "load" : [ {} ]
}
```
Features

- Self-service monitoring and visualization platform
- REST API
- Authentication
- Highly configurable
- Develop your own lenses (LDK)
## Refocus

Visualize service health

### Customers

<table>
<thead>
<tr>
<th>INSTANCE</th>
<th>Hooli</th>
<th>Planet Express</th>
<th>Cust3</th>
<th>Cust4</th>
<th>Cust5</th>
<th>Cust6</th>
<th>Cust7</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>APT</td>
<td>955.4</td>
<td>218.8</td>
<td>198</td>
<td>1.7K</td>
<td>0</td>
<td>343.7</td>
<td>166.3</td>
<td>0</td>
</tr>
<tr>
<td>CACHE</td>
<td>8.1</td>
<td>8.2</td>
<td>6.3</td>
<td>10.9</td>
<td>0</td>
<td>7</td>
<td>11.3</td>
<td>0</td>
</tr>
<tr>
<td>IO</td>
<td>4.7K</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.1K</td>
<td>11.9K</td>
<td>0</td>
</tr>
<tr>
<td>MEM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NET</td>
<td>43.6K</td>
<td>15.7K</td>
<td>8.8K</td>
<td>11.2K</td>
<td>0</td>
<td>20.4K</td>
<td>20.3K</td>
<td>0</td>
</tr>
<tr>
<td>LOAD</td>
<td>162K</td>
<td>16.7K</td>
<td>9.5K</td>
<td>16.7K</td>
<td>0</td>
<td>32.4K</td>
<td>71.5K</td>
<td>0</td>
</tr>
<tr>
<td>LAG</td>
<td>3.1K</td>
<td>7.3</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>101.7</td>
<td>0</td>
</tr>
<tr>
<td>ERROR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LIMIT</td>
<td>5.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12.7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LOGIN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PERF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>578</td>
<td>175</td>
<td>56</td>
<td>15</td>
<td>0</td>
<td>559</td>
<td>831</td>
<td>0</td>
</tr>
</tbody>
</table>
What tools do we use?
Open Source on Github - github.com/Salesforce

Argus (Java)
- Blog post
- GitHub.com/salesforce/Argus

Pyplyn (Java)
- API reference
- GitHub.com/salesforce/pyplyn

Refocus (Javascript)
- Blog post
- GitHub.com/Salesforce/refocus
All contributions are welcome!
Q&A
thank you