PUMP

Performance and Usage Monitoring Program

Demo of first functional implementation

21-March-2013

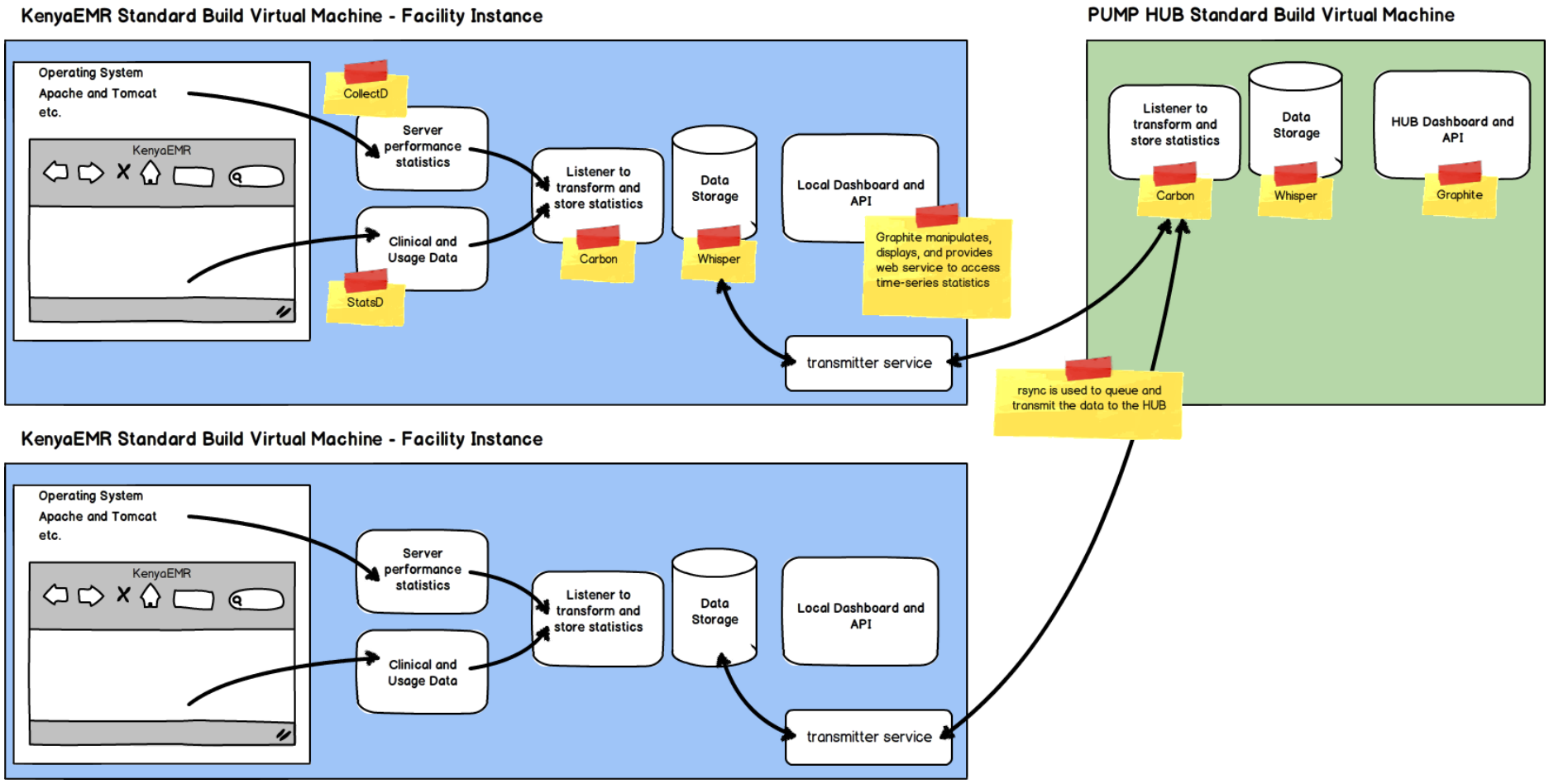
Jan Flowers, Bill Lober, Paul Bugni

Initial Functional Implementation

- collects system level statistics from the servers
  - **server performance indicators**
- collects application level statistics from the servers, including both
  - **usage indicators**
  - **clinical indicators**
- display those indicators locally on the KenyaEMR facility server
- push those indicators to a PUMP hub (consolidated statistics)
- display indicators from each server on the PUMP hub
- display combined indicators from several servers on the same graph on the PUMP hub

Full functional and technical specifications: http://goo.gl/owoUGe
PUMP Demo Doc: http://goo.gl/ezP93
KenyaEMR Indicators: http://goo.gl/g2wECA

What the Workflow Looks Like...



Current Live and Example Stats

System

- cpu-idle = indicator of how much cpu is being used
- memory-free = how much memory is free to use

Clinical

- # of patients with CD4 cell count in past 6 months
- # of HIV positive patients

Usage

- # logins
- # patient charts opened = # charts opened
- # retrospective patients entered into system = # MOH 257 formed saved / day
- # visits entered = # of patients checked-in

URLs for the Demo

Main System URLs

KenyaEMR system #2 SeattleEMR02-MFL_54321
https://69.91.227.144/openmrs/index.htm (u/p is admin/test for OpenMRS demo)

KenyaEMR system #1 SeattleEMR01-MFL_12345
http://69.91.227.159/ (server root - currently not configured)

PUMP centeral server (hub)
http://69.91.227.143/ (server root - currently redirects to Graphite)

PUMP display URLs

http://69.91.227.143/graphite (hub)
http://69.91.227.159/graphite (remote-1)
http://69.91.227.144/graphite (remote-2)

Custom Dashboard Example using Graphite

http://69.91.227.143/dashboard/#both_remotes (example dashboard)

69.91.227.143

cirg | software | development | informatics | news | blog | resources | iphone-bmw | design | research/grants | school

graphite

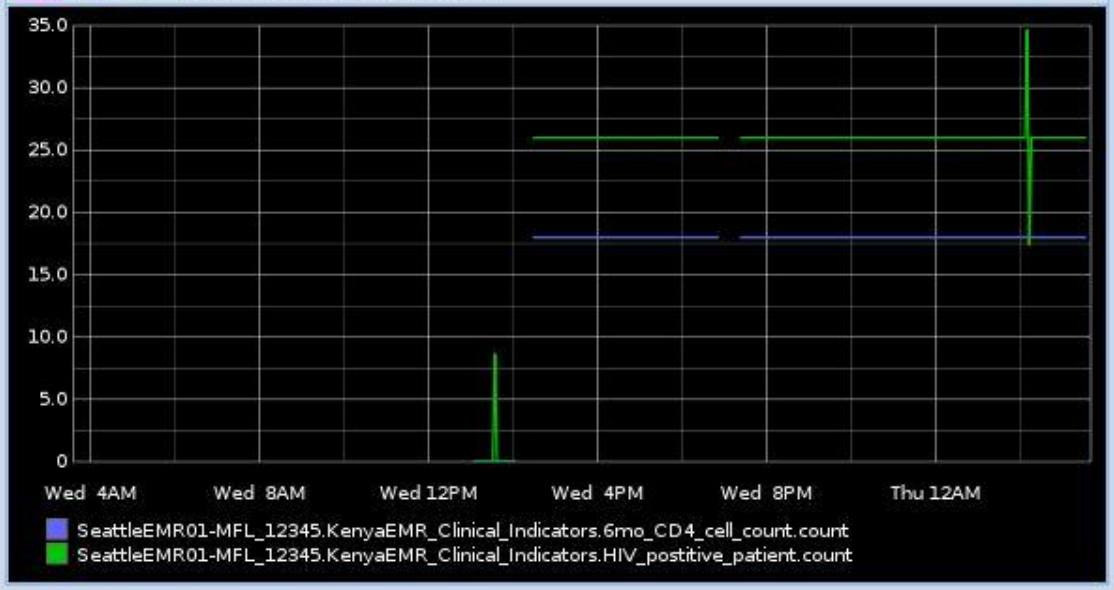
Login
Documentation
User Interface: Dashboard | flot (experimental) | events (experimental) |

Tree | Search | Auto-Completer

- Graphite
  - SeattleEMR01-MFL_12345
    - KenyaEMR_Clinical_Indicators
      - 6mo_CD4_cell_count
      - HIV_postitive_patient
    - KenyaEMR_Usage_Indicators
      - logins
      - patient_charts_opened
      - retrospective_patients_entered
      - visits_entered
    - Server_Performance
      - cpu-0
      - memory
    - statsd
  - SeattleEMR02-MFL_54321
- User Graphs

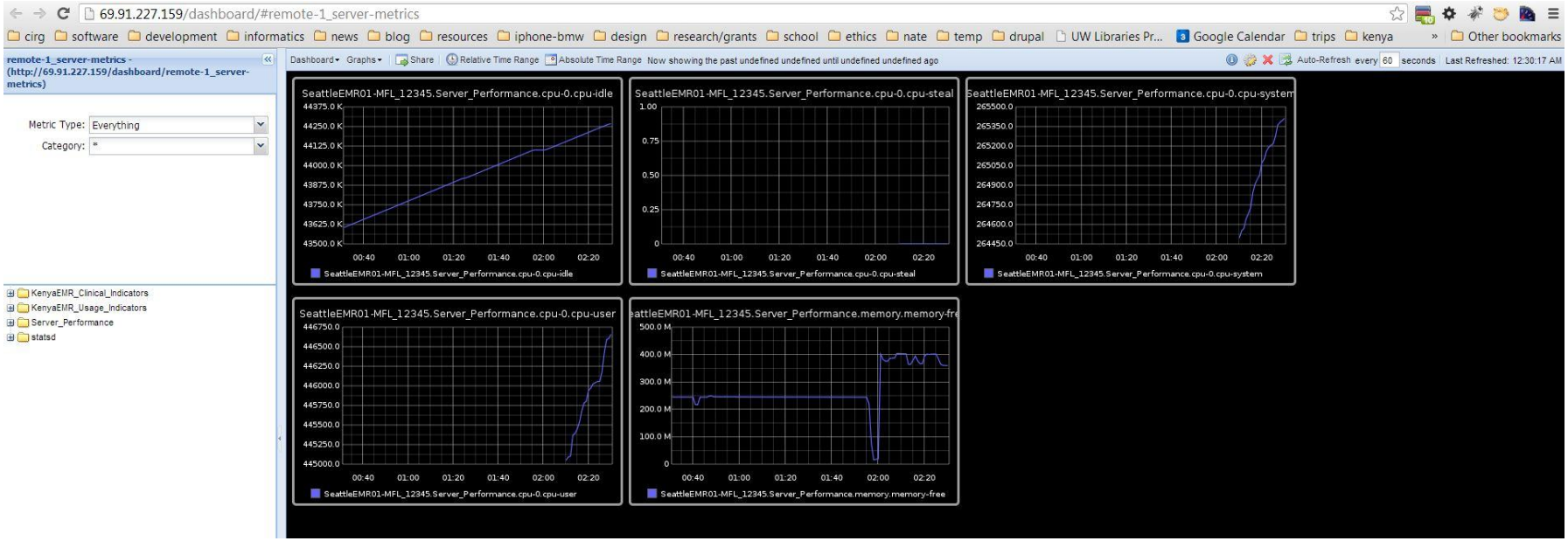
Graphite Composer

Now showing the past 24 hours

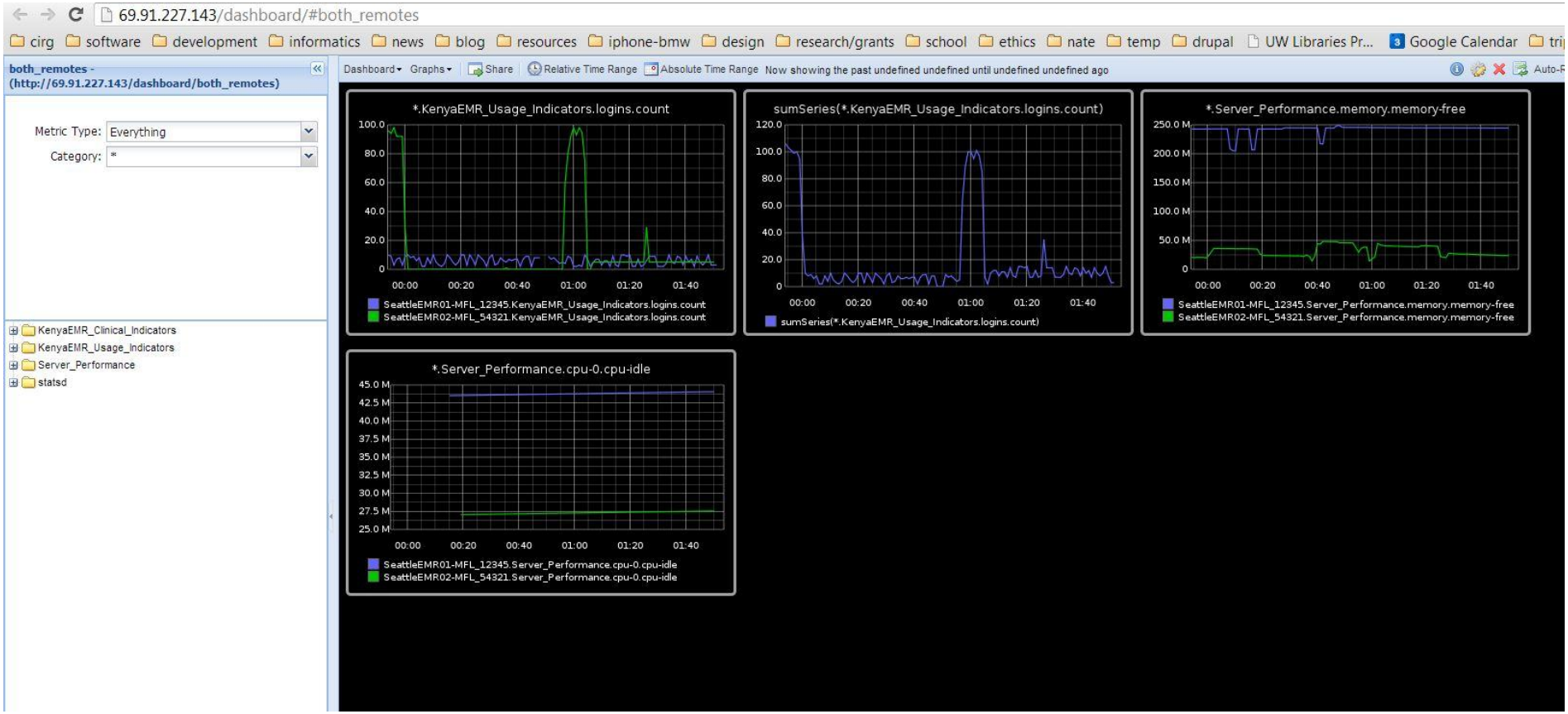


Graph Options | Graph Data | Auto-Refresh

System Performance Dashboard



HUB Combined Dashboard



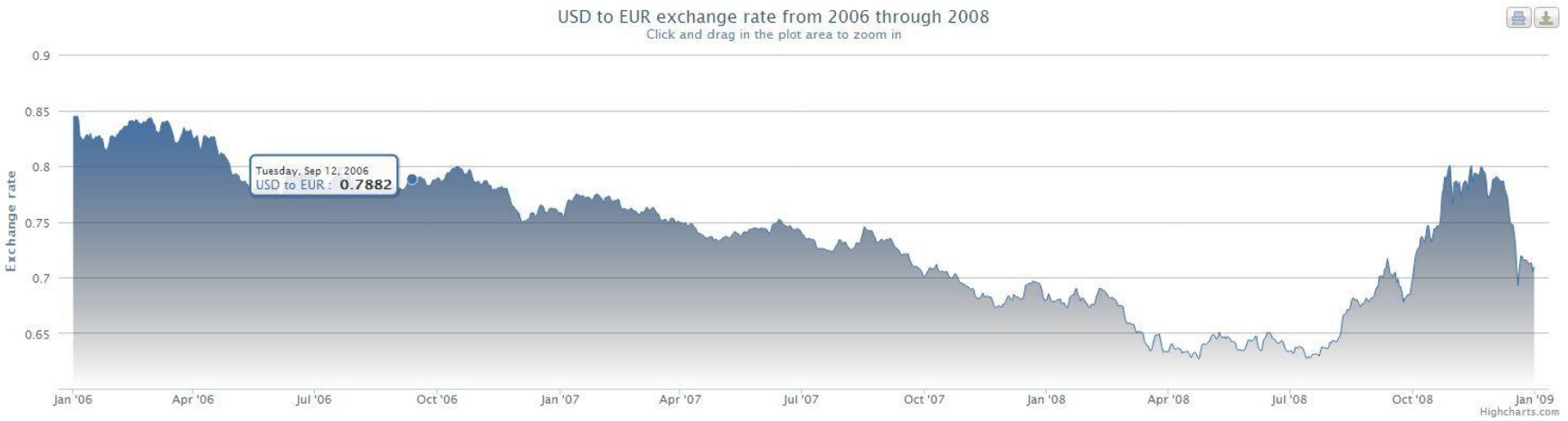
GeekSpeak

Built on open source technologies:

- statsd (python) - to collect data from applications (created by etsy)
- collectd - collects server level performance statistics
- carbon - daemon to listen and store data; helps to mitigate connectivity issues
- whisper - round-robin like database used by carbon to store time series data for display
- rsync - queues and transmits the data between the local servers and the hub
- graphite - graphic display of collected statistics/indicators (created by orbitz)

Improving the UI

HighCharts Dynamic Charting and Drill-Down Capacity



View options | Edit in jsFiddle | « Previous | Next »

Improving the UI

Next Steps

1. Need to configure graphite for SSL,
2. change DNS for pump.kenyaemr.org to point to .143 [11:58:08 AM] Bill Lober: actually, how about letting the machine know itself as pump-demo.ke.... and setting up that DNS, but setting up either a CNAME or a redirect to have pump.ke... point to that hub temporarily.
3. Kill existing pump.kenyaemr.org machine
4. Implement convention that the root of each machine is a page w links to whatever is interesting on that machine. Brand those root pages as either production or non production w/ a background color, perhaps.
5. replace this functional implementation with a newer functional implementation (version 2) using data from the dev, demo, and test servers, once PUMP is in a production release.
6. Deploy production hub for KE servers
7. Incorporate PUMP into KenyaEMR server 13.1.1, or the next release
8. for site-level statistics, should ideally add web-module login (which means adding web module auth to OpenMRS also)
9. currently there is no configurations steps for implementers, but we need to add the ability for implementers to configure which hub (or hubs) the local installation will send data to
10. county level vs. national level servers? (per Bill's discussion w/ Patrick - clarify this requirement, alternatives, and implications)
11. define a process for installing the rsync user's public key on the appropriate hub. When the puppet pump install script is run, a new ssh key pair is generated on the new remote system. This file </home/pump_rsync_user/.ssh/id_rsa.pub> must be appended to the pump_rsync_user's authorized_keys file </home/pump_rsync_user/.ssh/authorized_keys> on the correct hub.