title: Transaction Tracing description: Trace code execution from beginning to end, see insights into operations about various data stores with database operations report, and trace arbitrary applications, including backend and command-line apps, batch jobs like MapReduce and more

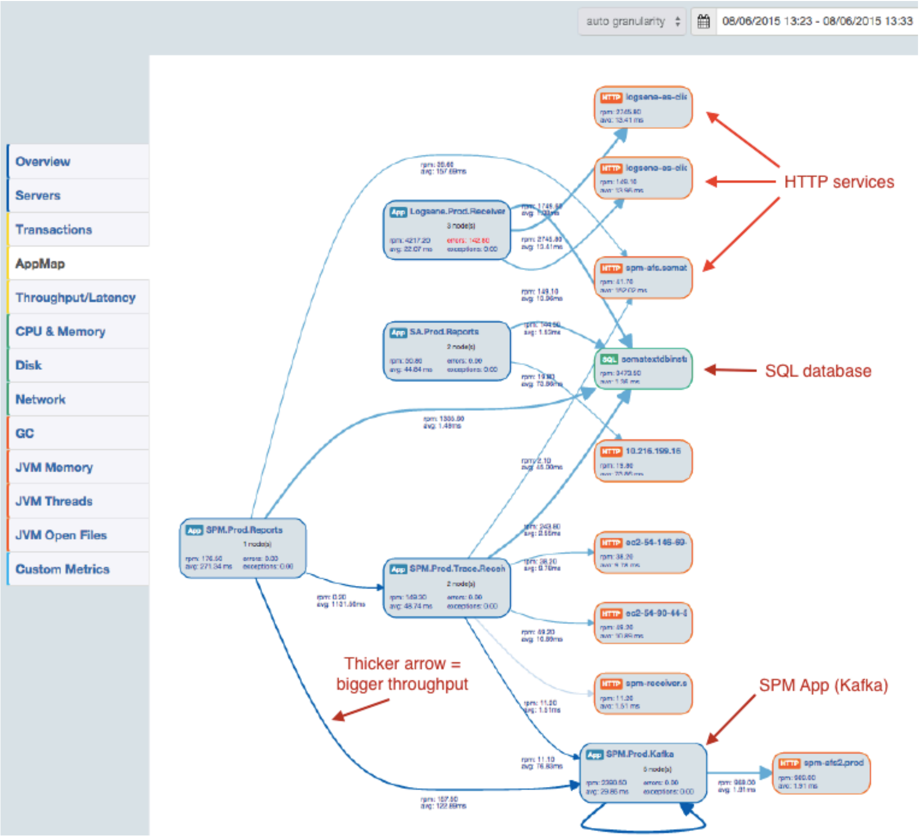
What is Transaction Tracing

Transaction tracing lets one trace code execution from beginning to end. It also enables Database Operations reporting. In case of Sematext Monitoring, transaction tracing can also cross applications, networks, and servers.

For example, you can trace code execution from a beginning of an HTTP request made to a web application through any calls this web application makes to relational, NoSQL, or any other databases or backend servers and services like Elasticsearch or Solr or Kafka, etc., all the way to the response the application returns to the original caller.

Transaction tracing is not limited to HTTP requests - one can also trace arbitrary applications, including backend apps, command-line apps, batch jobs like MapReduce, etc. Such tracing makes it possible to find performance bottlenecks in running code, whether in production or some other environment.

Starting with version 1.24.10, the Sematext Agent provides ability to track application transactions, thus making it easier to find bottleneck in running applications and troubleshoot performance issues.



Notes

- Transaction Tracing requires App Agent running in embedded mode.
- Enabling/disabling the tracing agent requires Sematext Agent restart, which means it requires the restart of the application running the embedded App Agent.
- Enabling transaction traces adds only about 1% CPU overhead.
- Transaction Tracing is different from On Demand Profiling.

Resources

- http://blog.sematext.com/blog/transaction-tracing-performance-monitoring/
- http://blog.sematext.com/blog/transaction-tracing-reports/
- http://blog.sematext.com/blog/introducing-appmap/