NATION

User Conference

OCTOBER 24-26

2017

Jesse Dunn
Strategic Technical Account Manager
Jamf

Garett Klingbeil
Sr. Technical Support Engineer
Jamf

Doug Worley
Sr. Professional Services Engineer
Jamf

Joshua Roskos
Professional Services Engineer
Jamf

Tomcat & MySQL Tuning

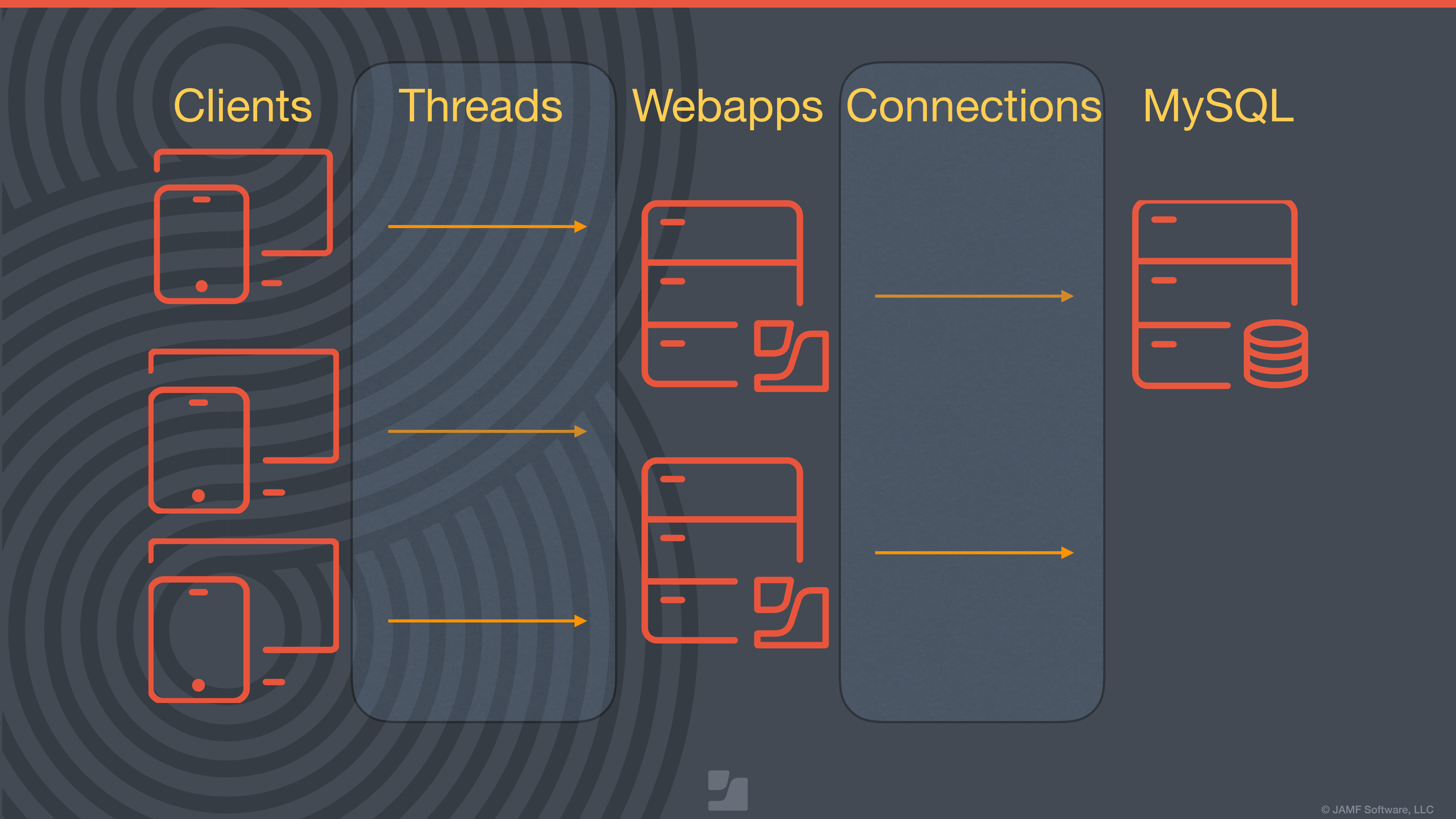
Presentation agenda:

The Basics

Tomcat Configuration

MySQL Configuration

The Basics



Data Flow

- Devices generate a thread for every task
- Threads hang out in maxThreads Pool
- As threads connect, Tomcat leverages both CPU and RAM.

Data Flow

- Thread —> Connection
- MySQL processes each connection as a query, and returns the data to Tomcat
- Tomcat processes the return.

Hardware and Process Tuning

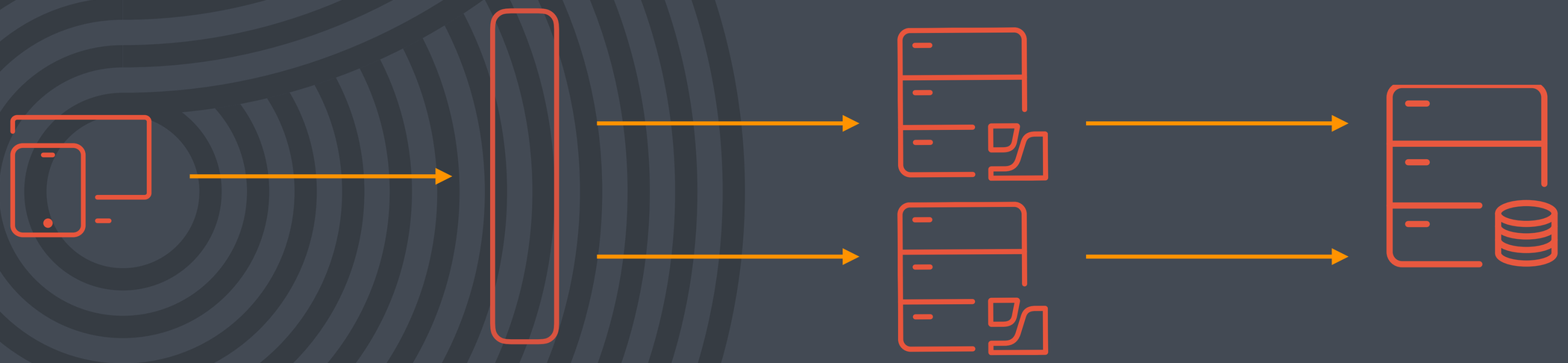
- Dedicated Virtual Machines are Amazing!
- High Performance: 50-75% to Tomcat/MySQL
- No Magic Equation
- Plan for the worst

Adjusting Tomcat Settings

- JN Article *“Allocating Additional Memory to Tomcat”*
  - *https://jamf.it/kfVRM*
- macOS
  - /Library/LaunchDaemons/com.jamfsoftware.tomcat.plist
- Windows
  - C:\Program Files\JSS\Tomcat\bin\tomcat8w.exe
- Linux
  - /usr/local/jss/tomcat/bin/setenv.sh

Clustering - Adding Jamf Pro Webapps

- Adding a new Tomcat Webapp (Clustering) allows multiple servers to talk to the database
- Don’t forget the Load Balancer



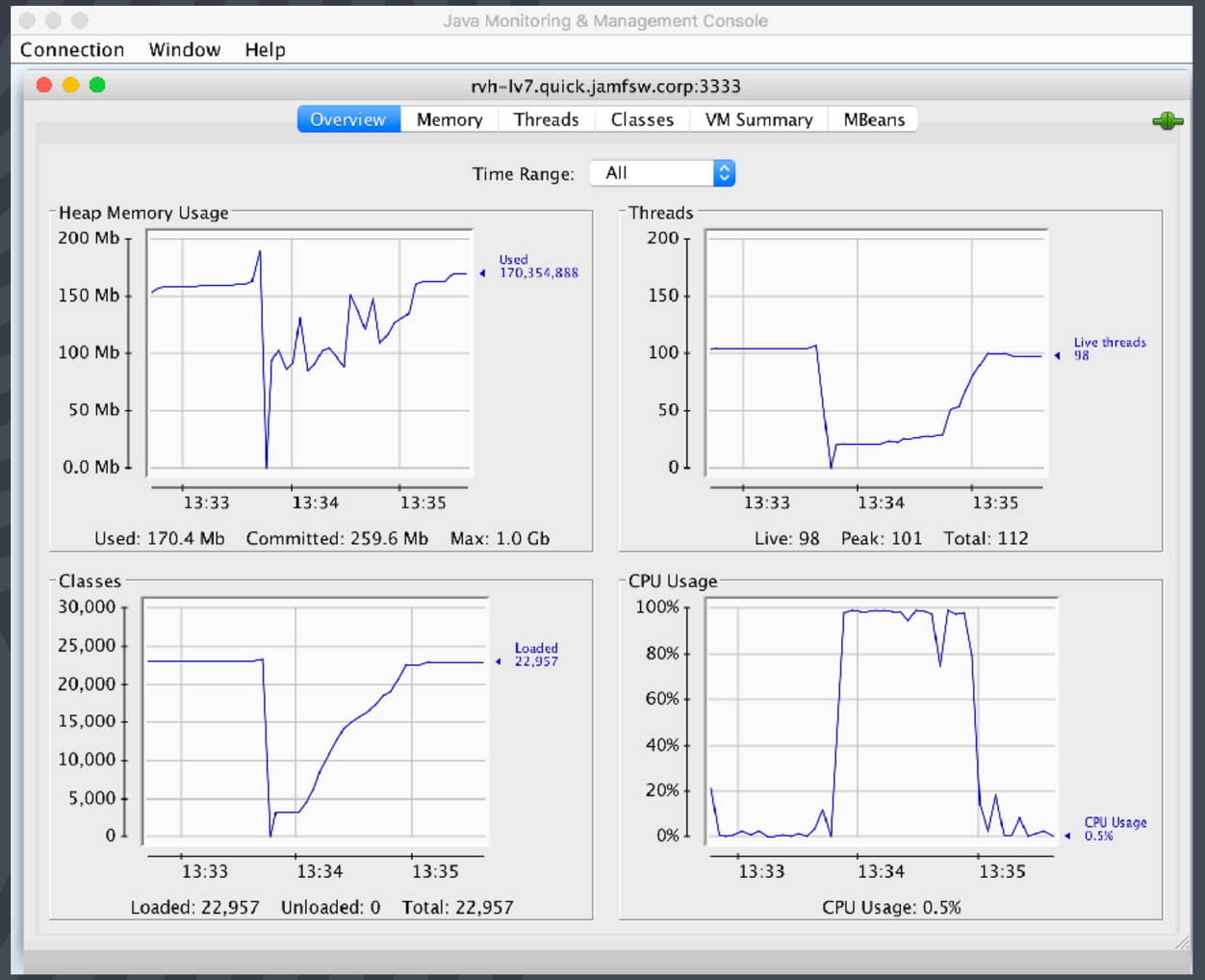
Trending & Monitoring Software

- JConsole
- Zabbix
- Third Party Applications

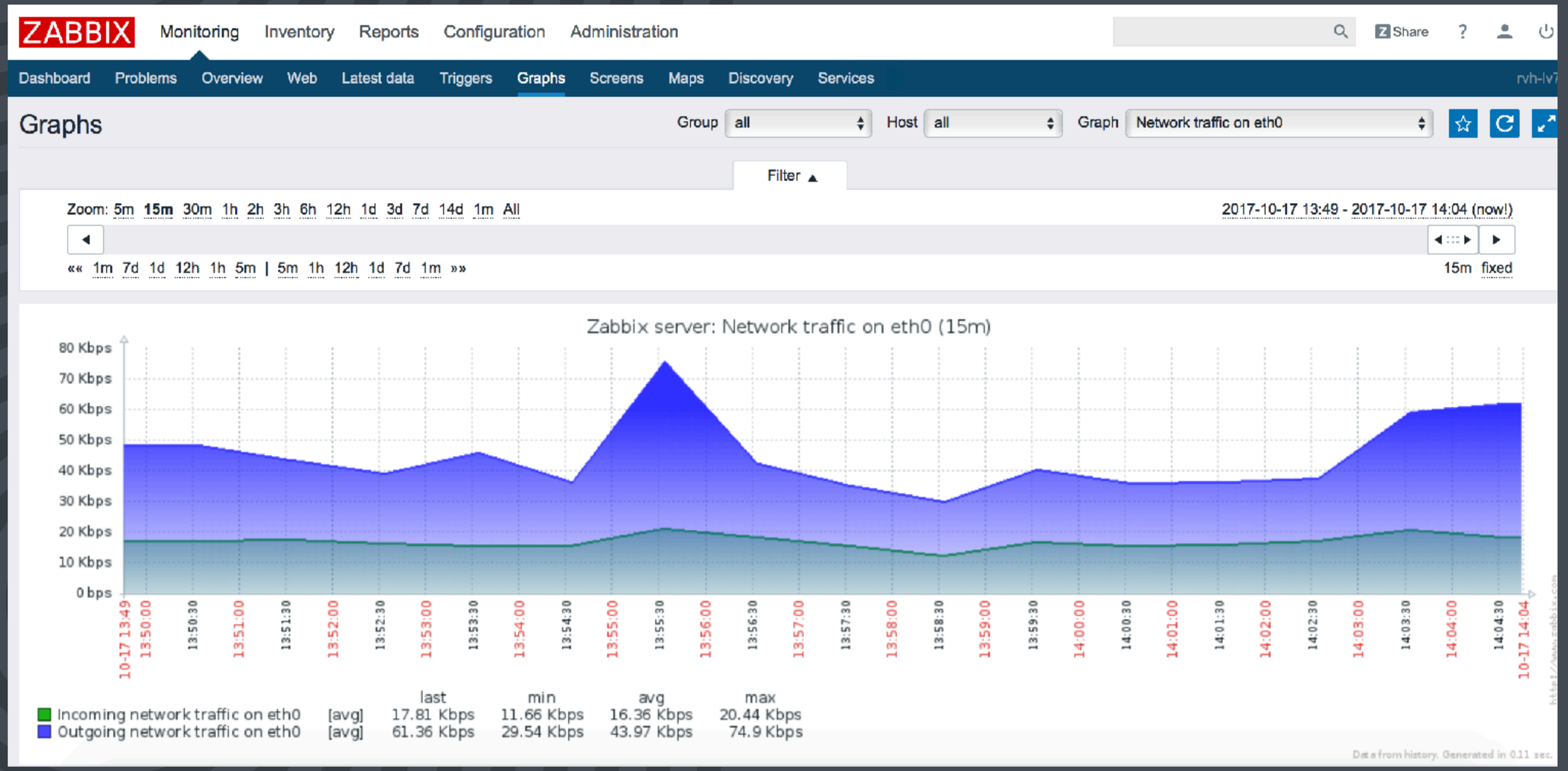
Trending & Monitoring Software

- Enable JMX monitoring in setenv.sh / tomcat8w.exe / com.jamfsoftware.tomcat.plist
- Connect Monitoring application

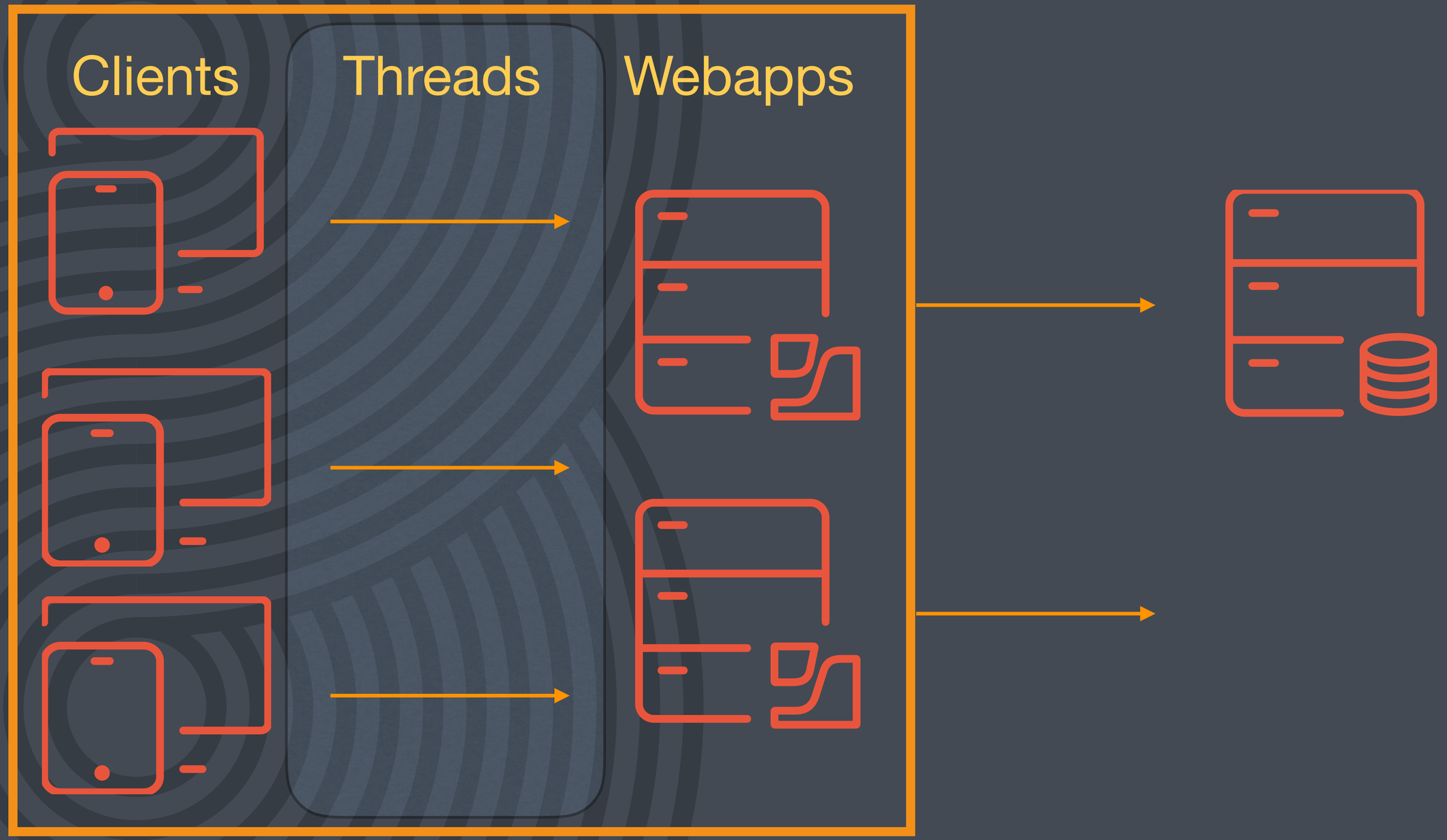
Trending & Monitoring Software



Trending & Monitoring Software



Tomcat Configuration



Server.xml

```
Location: /path/to/jss/tomcat/conf/server.xml
Contents of file:

<Executor name="TomcatThreadPool" namePrefix="Tomcat-" maxThreads="200" minSpareThreads="4" />

<Connector URIEncoding="UTF-8" port="8080" executor="TomcatThreadPool" protocol="HTTP/1.1"
connectionTimeout="20000" redirectPort="8443" proxyPort="443" scheme="https" />

<Connector URIEncoding="UTF-8" server="Apache" port="8443" executor="tomcatThreadPool"
SSLEnabled="true" maxPostSize="-1" scheme="https" protocol="HTTP/1.1" secure="true"
clientAuth="false" sslProtocol="TLS" sslEnabledProtocols="TLSv1.2,TLSv1.1,TLSv1"
keystoreFile="/Library/JSS/Tomcat/TomcatSSLKeystore" keystorePass="*****"
ciphers="[truncated]"> <!--keystoreFile updated by JSS. Wed Oct 05 11:30:00 CEST 2016
```

maxThreads

The larger the maxThread pool, the more simultaneous connections

- Allows more devices to connect at the same time
- Can be spread out to multiple Webapps

As threads connect, Tomcat pulls more CPU and RAM

- More simultaneous connections require additional resources to process
- Tomcat handles serial connections faster than parallel connections

Pipe analogy or diagram?

maxThreads - Stress Points

High CPU/RAM can indicate too-high maxThreads

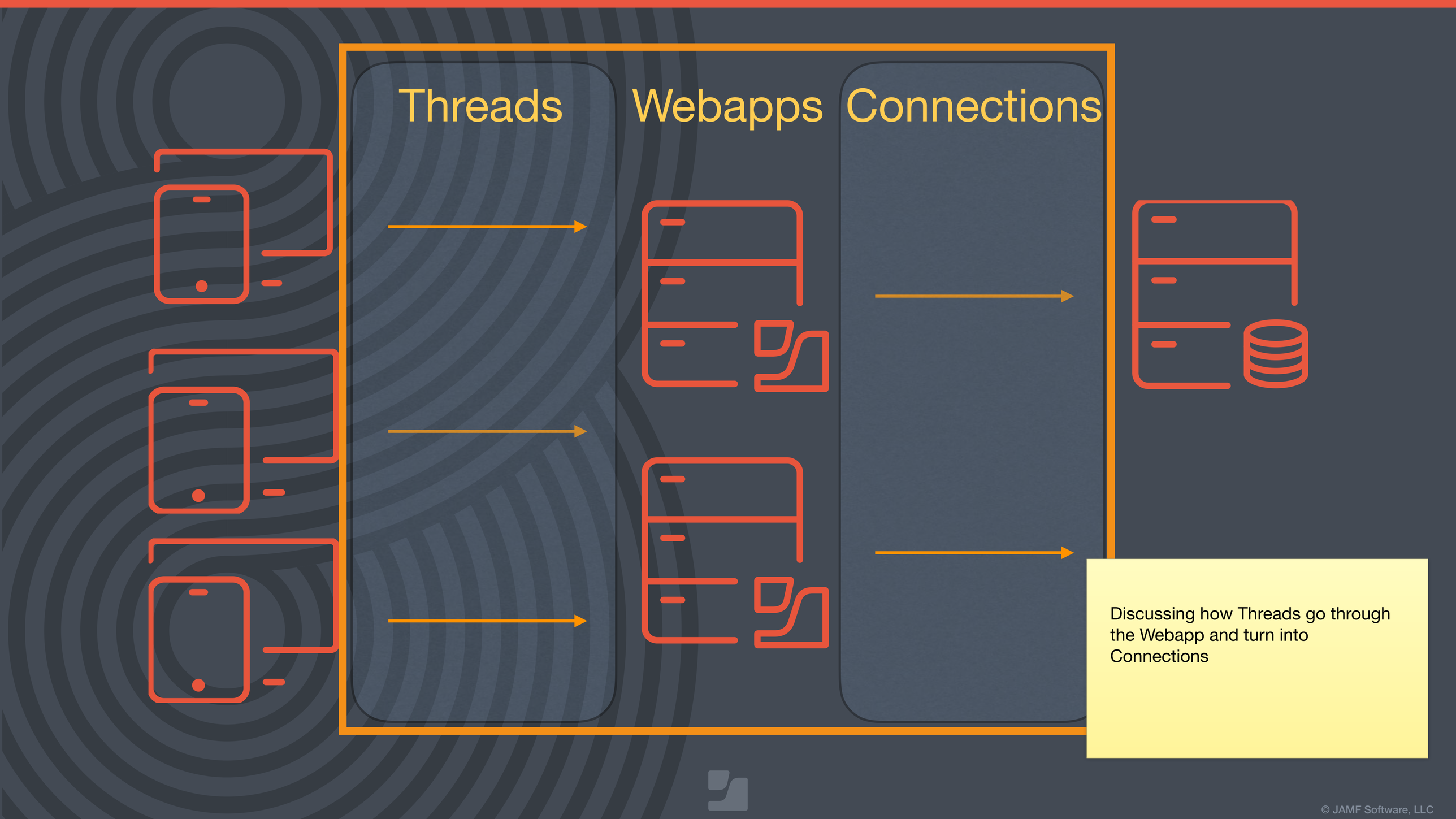
- Balancing Act between CPU/RAM and necessary threads
- Ideal usage is 50%-70%

Does not always indicate scaling is incorrect

- Long running SQL Queries can spike Tomcat
- Multiple factors in play - Server hardware, network connectivity, etc..

maxThreads - Starting Point

- Defaults to 200 - current recommendation
- Does Jamf Pro remain responsive?
- CPU and RAM at acceptable levels?
- Then - increase or decrease
- Monitoring tools are the key to scaling



Discussing how Threads go through the Webapp and turn into Connections

Database.xml

```
Location: /path/to/jss/tomcat/webapps/ROOT/WEB-INF/xml/DataBase.xml
Contents of file:

<DataBase>
    <DataBaseType>mysql</DataBaseType>
    <DataBaseDriver>org.mariadb.jdbc.Driver</DataBaseDriver>
    <ServerName>localhost</ServerName>
    <ServerPort>3306</ServerPort>
    <DataBaseName>jamfsoftware</DataBaseName>
    <DataBaseUser>jamfsoftware</DataBaseUser>
    <DataBasePassword>**********</DataBasePassword>
    <MinPoolSize>5</MinPoolSize>
    <MaxPoolSize>45</MaxPoolSize>
    <MaxIdleTimeExcessConnectionsInMinutes>1</MaxIdleTimeExcessConnectionsInMinutes>
    <MaxConnectionAgeInMinutes>5</MaxConnectionAgeInMinutes>
    <NumHelperThreads>3</NumHelperThreads>
    <InStatementBatchSize>1000</InStatementBatchSize>
    <jdbcParameters>?characterEncoding=utf8&amp;useUnicode=true&amp;jdbcCompliantTruncation=false</jdbcParameters>
    <EnableJMX>true</EnableJMX>
</DataBase>
```

maxPoolSize

Defines how many simultaneous connections can be open between Tomcat and MySQL

- Previous default was 90
- Current default is 45

MySQL handles serial operations faster than parallel operations

- Allows more CPU Power per query
- Increases Tomcat-to-MySQL throughput

maxThread “pool” means we can have more threads than connections

maxPoolSize - Stress Points

Monitoring

- MySQL Process List shows the command type, the time of life, and current state
- Analyzing this output helps determine beneficial my.cnf/my.ini modifications

Proclist Monitor

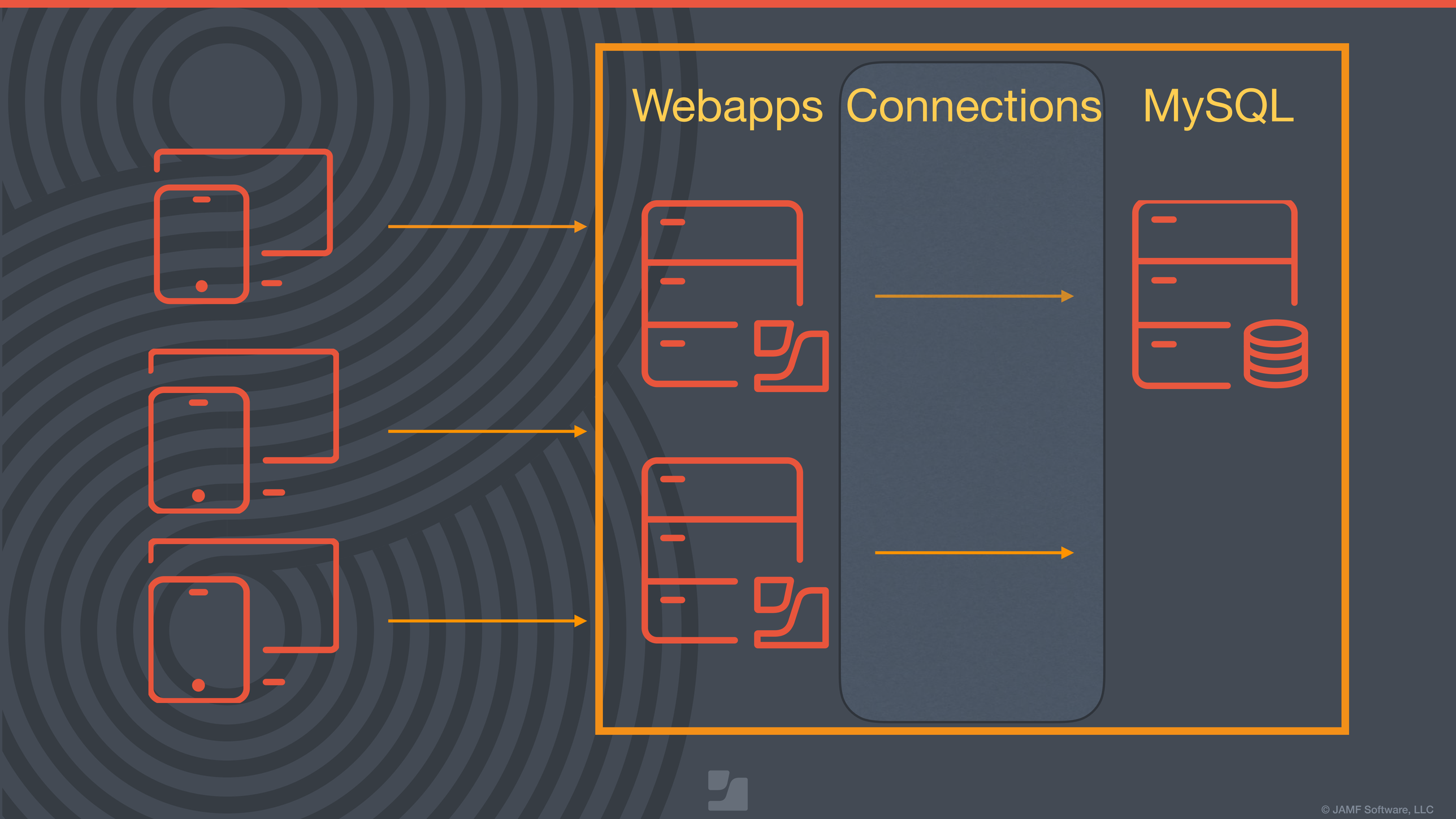
- A script to export the process list to a log
- Can run for scheduled periods for monitoring

maxPoolSize - Starting Point

Finding the maxPoolSize Balance

- There’s no exact answer - every environment is different
- Need to balance throughput with availability
- The new default of 45 is a good place to start
- If we see connections take longer than “0” to process, can scale that back on each of the Tomcat node
- Number of Webapps plays a factor

MySQL Configuration



my.cnf/my.ini

```
Location: Varies based on Operating System and MySQL Version
Contents of File:  Depends on Operating System

# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html


!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
[mysqld]
max_allowed_packet=2048M
max_connections=80
general_log=0
general_log_file=/tmp/generalquery.log
open_files_limit = 5000
```

Linux Example

my.cnf/my.ini

Command Line vs my.cnf/my.ini

- Settings modified via MySQL Command Line Interface are only present until MySQL Service is restarted
- Settings in my.cnf/my.ini are applied during MySQL Service initialization
- If values are undefined, MySQL uses defaults
- Insert under [mysqld] unless otherwise specified
- Windows installations have many variables already preconfigured

my.cnf/my.ini - Stress Points

Any unexpected behavior with MySQL could be caused by the config

- Monitor the process list, CPU, and RAM
- Multiple variables that work together
- Spikes in MySQL CPU/RAM can bleed into Tomcat - Start by examining MySQL

Process List Output (show processlist)

```
| 39392 | jamfsoftware | 10.11.12.13:49958 | jamfsoftware | Query   | 0    | Sending data
| 39393 | jamfsoftware | 10.11.12.13:49959 | jamfsoftware | Query   | 0    | statistics
| 39394 | jamfsoftware | 10.11.12.13:49960 | jamfsoftware | Query   | 0    | statistics
| 39395 | jamfsoftware | 10.11.12.13:49961 | jamfsoftware | Query   | 0    | Sending data
| 39396 | jamfsoftware | 10.11.12.13:49962 | jamfsoftware | Query   | 0    | statistics
| 39397 | jamfsoftware | 10.11.12.13:49963 | jamfsoftware | Query   | 0    | statistics
```

```
| 39392 | jamfsoftware | 10.11.12.13:49958 | jamfsoftware | Query   | 0    | Sending data
| SELECT mobile_devices.mobile_device_id AS mobile_device_id, mobile_devices.udid AS udid, mobile_devi |
| 39393 | jamfsoftware | 10.11.12.13:49959 | jamfsoftware | Query   | 0    | statistics
| SELECT mobile_devices.mobile_device_id AS mobile_device_id, mobile_devices.udid AS udid, mobile_devi |
| 39394 | jamfsoftware | 10.11.12.13:49960 | jamfsoftware | Query   | 0    | statistics
| SELECT mobile_devices.mobile_device_id AS mobile_device_id, mobile_devices.udid AS udid, mobile_devi |
| 39395 | jamfsoftware | 10.11.12.13:49961 | jamfsoftware | Query   | 0    | Sending data
| SELECT mobile_devices.mobile_device_id AS mobile_device_id, mobile_devices.udid AS udid, mobile_devi |
| 39396 | jamfsoftware | 10.11.12.13:49962 | jamfsoftware | Query   | 0    | statistics
| SELECT mobile_devices.mobile_device_id AS mobile_device_id, mobile_devices.udid AS udid, mobile_devi |
| 39397 | jamfsoftware | 10.11.12.13:49963 | jamfsoftware | Query   | 0    | statistics
| SELECT * FROM mobile_device_configuration_profile_deployment WHERE mobile_device_configuration_profi |
```

my.cnf/my.ini - Starting Point

Common Modifications by Support

- max_connections
- optimizer_search_depth
- query_cache_type
- table_open_cache
- key_buffer_size

max_connections

Maximum number of connections MySQL will allow

- Needs to equal or exceed the total number of maxPoolSize incoming connections, +1 for a connection outside the Webapps
- Just a ceiling - setting this higher than necessary is a possibility.

optimizer_search_depth

Optimizer Search Depth handles the “stastics” portion of the SQL query.

- Depending on the complexity (number of tables affected by joins, number of READs on columns) it can greatly effect the time evaluating the optimal execution plan
- Defaults to 64. Support generally uses 3.

query_cache_type

MySQL's built in tool for caching queries for quick retrieval

- Useful for environments where the same query runs multiple times
- Jamf Pro environments use unique queries almost every time
- Set to 0 to disable completely
- If disabling, set query_cache_size to 0 as well

table_open_cache

Number of tables MySQL has cached for retrieving

- If the number of open tables is increasing rapidly we can increase this number
- Found with “show global status like ‘opened_tables’;”
- Can see number of currently open tables with “select @@table_open_cache”
- Generally we update by segments of 2000 (2000 to 4000 to 6000) until we find the optimal number
- Some OS’s limit table_open_cache increases, and modified values will not take effect

key_buffer_size

Determines the size of index buffers held in memory, which plays into how fast MySQL reads from the database

- Rule of thumb is about 25% of Server RAM
- Maxes out at 4 GB

Questions?

Thank you!