MAVEN

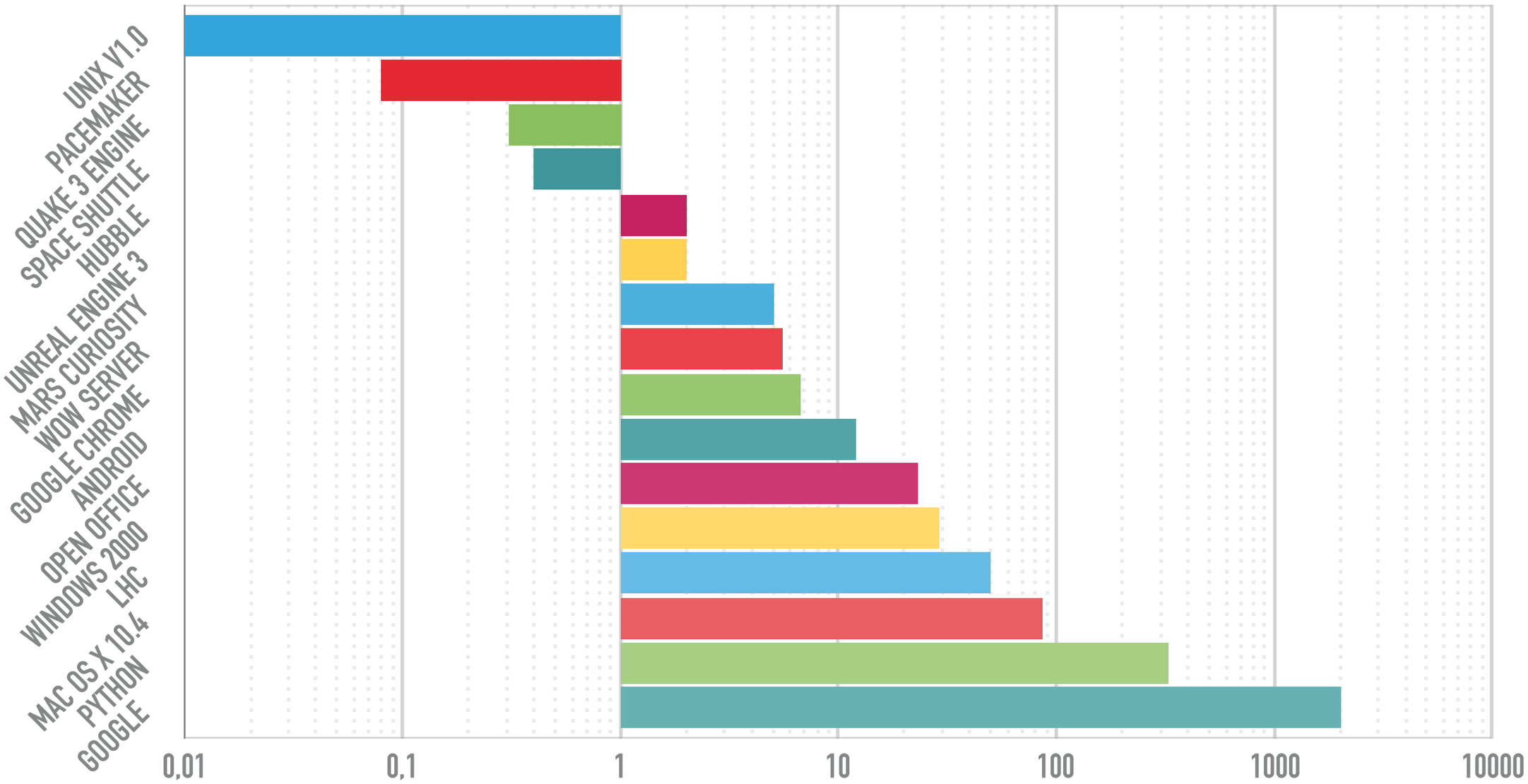
M.Barsacchi

AGENDA

- WHAT IS MAVEN
- THE POM
- LIFECYCLES
- ARCHETYPES
- … AN EXAMPLE …

BUILD TOOL

Build tools automate creation of executable application from sources. Building incorporate compiling, linking, packaging the code into a usable or executable form.



WHAT IS MAVEN?

- Maven comes from a Yiddish word, meaning *accumulator of knowledge*
- *Maven is a*
  BUILD TOOL
  DEPENDENCY MANAGEMENT TOOL
  DOCUMENTATION TOOL
- *Formally is a project management tool, which encompasses a project object model, a set of standards, a project lifecycle, a dependency management system and logic for executing plugin goals at defined phases in a lifecycle.*

WHY MAVEN?

- Project Oriented
- Maven tracks and downloads all the dependencies avoiding *"dependency hell"*
- *"Convention over Configuration"* (standardised project layout)
- Common interface for building software
- Tons of nice plug-ins

STANDARDISED DIRECTORY LAYOUT

```
PROJECTDIR/
        POM.XML
        SRC/
                MAIN/
                        JAVA/
                                ...
                        RESOURCES/
                                META-INF/
                                        SPRING/
                                                *.XML
                        OSGI-INF/
                                BLUEPRINT/
                                        *.XML
                TEST/
                        JAVA/
                        RESOURCES/
        TARGET/
                ...
```

- POM contains a complete description of how to build the project
- src directory contains all of the source material for building the project, its site and so on.
- target directory contains the results of the build (typically a JAR), as well as all the intermediate files.

N.B. While it is possible to override the standard directory layout, this is not a recommended practice in Maven.

THE POM

- Maven is based on the concept of project object model (POM)
- XML file, always residing in the base directory of the project as pom.xml. Users defined POMs extend the Super POM.
- The POM contains information about the project and various configuration detail used by Maven to build the project.
- The Maven POM is **declarative**,
  no procedural details are needed.

POM STRUCTURE

The POM contains four categories of description and configuration:

- General project information
  human readable informations
- Build Settings
  add plugins, attach plugin goal to lifecycle
- Build Environment
  describe the “family” environment in which Maven lives
- POM Relationships
  coordinates, inheritance, aggregations, dependencies

AN EXAMPLE OF POM

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.sparkexamples</groupId>
    <artifactId>first-example</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_2.10</artifactId>
            <version>1.2.0</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.1</version>
            </plugin>
        </plugins>
    </build>
</project>
```

TWO WORDS ABOUT VERSION

- Coordinates define the unique place of the project in the Maven universe. They are made up of <groupID>, <artifactID> and <version> *(The Maven trinity!)*
- Project versions are used to group and order releases:
  <major version>.<minor version>.<incremental version>-<qualifier>
  es: 1.2.3-alpha-2
- If the qualifier contains the keyword -SNAPSHOT, Maven will expand this token to a date and time value converted to UTC.

DEPENDANCY SCOPE

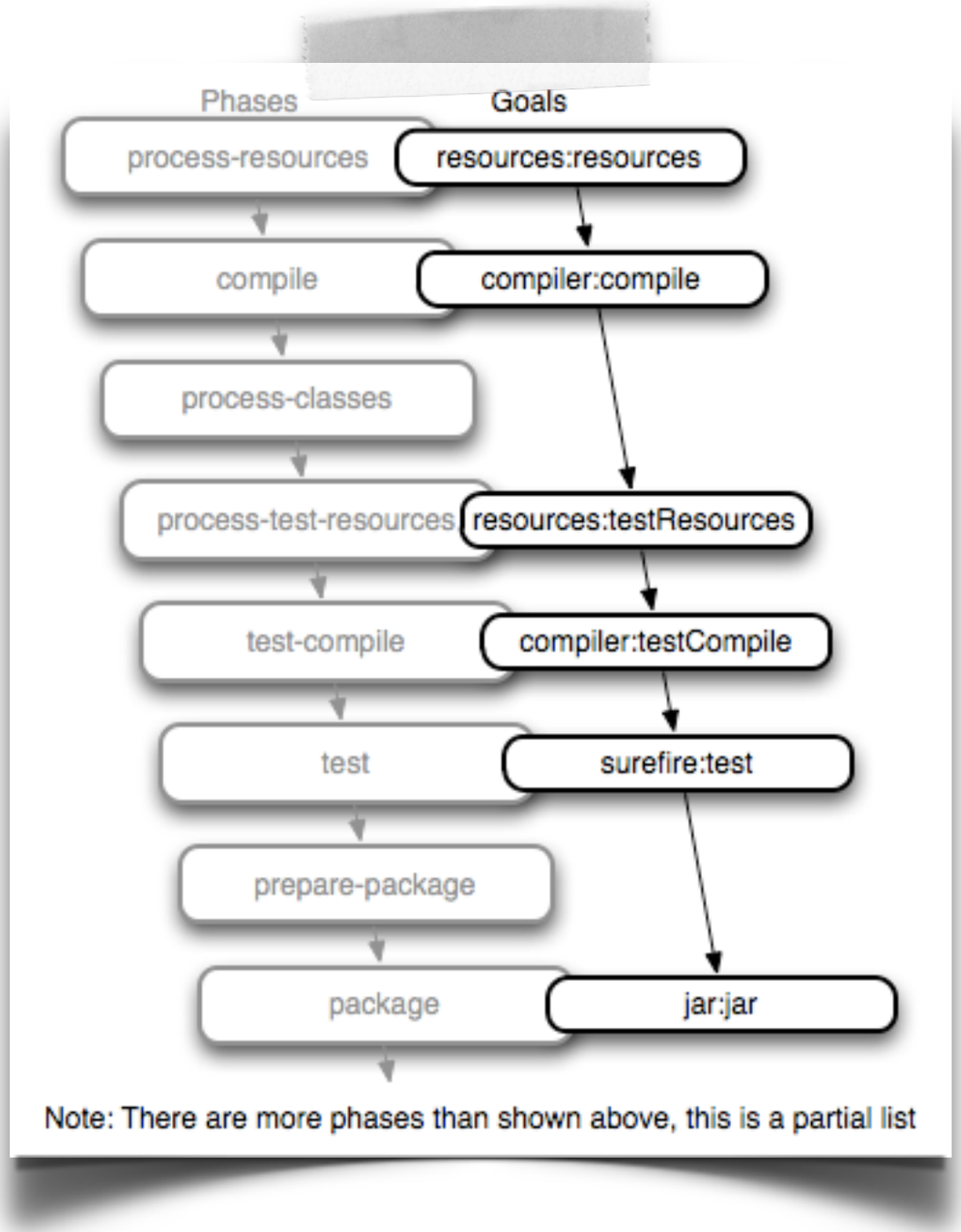
- compile, is the default scope; available in all classpath, they are packaged.
- provided, are available on the compilation classpath, but not at runtime. They are not transitive nor they are packaged.
- runtime, dependencies required for execute and test but not for compilation.
- test, test-scoped dependencies only.
- system, like provided, but you have to explicitly provide a path to the JAR on the local file system.

MAVEN LIFECYCLE

- A lifecycle is a organised sequence of phases that give order to a sequence of goals.
- Goals are packaged in plugins that are tied to phases.
- Three Standard Lifecycles:
  - Clean
  - default (or build)
  - Site

LIFECYCLES

- Lifecycles are made up of Phases
- and Phases are made up of Plugin Goals, which are specific tasks.
- Lifecycles phases are sequentially executed to complete a cycle.
- Calling a specific phase in a build cycle will trigger every prior build phase.

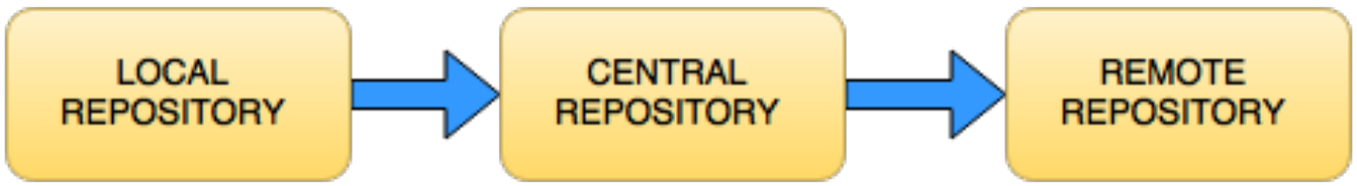


MAVEN REPOSITORY

- Directory of packaged JAR files with pom.xml.
  Maven searches for dependencies in the repositories.
  Three types of maven repository exists:

  - Local Repository
  - Central Repository
  - Remote Repository

REPOSITORY SEARCH ORDER



PLUGINS

- Plugins are the core feature of Maven;
  most of the action in Maven happens in plugin goals.
- Plugins affect Maven Lifecycle and offer access to goals.
- Plugins provide goals, and can be executed by following:

```
mvn [plugin-name]:[goal-name]
```

- Plugins take care of
  - Compiling sources
  - Packaging bytecode
  - Publishing sites
  - ……

ARCHETYPES

… an original pattern or model from which all other things of the same kind are made …

- An archetype is a template which is used by Maven Archetype plugin to create new projects.
- Artifacts in the repository can be used, creating a project based on an Archetype

```
mvn archetype:generate
```

- Develop quickly and consistent with best practices of your project.
- Tons of Archetype are provided in maven repository
  https://maven.apache.org/archetypes/index.html

GENERATING DOCUMENTATION

- Software applications are usually produced by team of developers.
- A well written documentation for a widely-distributed collection of users and developers is necessary.
- Maven can create project web site, as well as report on test failure or code quality.
- Site documentation can be easily generated

  mvn site

MAVEN INTEGRATION

- Maven is very well integrated in all popular IDEs for Java platform:
  - NetBeans 6.7+ - http://wiki.netbeans.org/Maven
  - Eclipse 3.x - http://www.eclipse.org/m2e/
  - IntelliJ IDEA - https://www.jetbrains.com/idea/help/maven.html

LET'S PLAY

NetBeans

LINEAR ALGEBRA WITH LA4J


```
<dependency>
  <groupId>org.la4j</groupId>
  <artifactId>la4j</artifactId>
  <version>0.5.5</version>
</dependency>
```

Eigen Decomposition

$$A = PDP^{-1}$$

If X is a real valued non singular matrix, will be positive definite

$$A = X^T X$$

N.B. A must be diagonalisable, i.e. if A is a nxn matrix it should be eigendecomposable in n eigenvectors.

```
import org.la4j.LinearAlgebra;
import org.la4j.Matrix;
import org.la4j.decomposition.MatrixDecompositor;
import java.utils.Random;

...
Random random = new Random();
Matrix a = new Matrix.random(n,n,random);

Matrix b = a.multiplyByItsTranspose();

MatrixDecompositor decompositor = b. withDecompositor(LinearAlgebra.EIGEN);
Matrix[] decomp = decompositor.decompose();

...
```

BIBLIOGRAPHY

- OFFICIAL MAVEN DOCUMENTATION - https://maven.apache.org/guides/
- MAVEN: THE COMPLETE REFERENCE - Sonatype - Tim O'Brien Manfred Moser John Casey Brian Fox Jason Van Zyl Eric Redmond Larry Shatzer - http://books.sonatype.com/mvnref-book/reference/index.html
- MAVEN BY EXAMPLE - Sonatype - Tim O'Brien John Casey Brian Fox Jason Van Zyl Juven Xu Thomas Locher Dan Fabulich Eric Redmond Bruce Snyder - http://books.sonatype.com/mvnex-book/reference/public-book.html
- NETBEANS MAVEN TUTORIAL - https://platform.netbeans.org/tutorials/nbm-maven-quickstart.html
- THE POM DEMYSTIFIED - http://www.javaworld.com/article/2071772/java-app-dev/the-maven-2-pom-demystified.html?page=1