Resource Sizing: Spotfire for AWS

With TIBCO Spotfire® for AWS, you can have the best in analytics software available at your fingertips in just a few clicks. On a single Amazon Machine Image (AMI), you get a multi-user platform with a variety of services preinstalled and preconfigured so you can:

- Connect to dozens of data sources, cloud and on-premises
- Explore your data through best-of-breed, interactive visual analytics
- Empower others in your organization by authoring and sharing dashboards and guided analytics

The purpose of this guide is to help you answer these questions:

- How do I determine which Amazon EC2 instance type to choose?
- When should I increase my capacity, and should I scale up or out?
- How do I deploy Spotfire for high availability?

SINGLE-AMI CONFIGURATION

The basic recommended AWS configuration is shown in Figure 1. Here an Amazon VPC has been created to host a single Spotfire instance for AWS EC2 running all the Spotfire components, plus the required Microsoft SQL Server instance.

To allow users to access the Spotfire instance via a fixed address, an elastic IP address is associated with instances and not attached to a virtual private cloud (VPC). An AWS security group with the Spotfire recommended port settings is also used to ensure that only the required ports are opened.
### DATA ACCESS AND PERFORMANCE IMPLICATIONS

You can access data with Spotfire in two primary ways:

<table>
<thead>
<tr>
<th>DATA ACCESS METHOD</th>
<th>DESCRIPTION</th>
<th>MAX DATA SIZE</th>
<th>BENEFITS</th>
<th>CONSIDERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-memory</td>
<td>Load all or subsets of the data into the Spotfire in-memory data engine. Spotfire performs all calculations.</td>
<td>&gt; 100M rows Based on system memory available.</td>
<td>• Access to an extensive set of transformations and calculations built into Spotfire • Unrivaled performance for ad-hoc analysis</td>
<td>• The load times will depend upon the speed of the network connection to the data source. • Once loaded into memory, some reasonable estimates of performance can be made. See following table.</td>
</tr>
<tr>
<td>In-database</td>
<td>Leave the data in the underlying database. Spotfire issues dynamic queries with every user interaction, bringing back only a small set of aggregated results required to refresh the visualizations.</td>
<td>Unlimited Based on scale of underlying database.</td>
<td>• Scales to very large data sizes, beyond what can fit in memory. • No up-front load times.</td>
<td>• Limited to the transformations and calculations supported by the underlying database. • Performance depends on the response of the underlying database, with some minimal overhead of the network connection.</td>
</tr>
</tbody>
</table>
PERFORMANCE AND SIZING GUIDELINES
Approximate sizing guidelines for the different instance types:

<table>
<thead>
<tr>
<th>INSTANCE TYPE</th>
<th>VCPUS</th>
<th>MEMORY (GB)</th>
<th>SMALL DATA</th>
<th>MEDIUM DATA</th>
<th>LARGE DATA</th>
<th>W/DATA IN DATABASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>m4.large</td>
<td>2</td>
<td>8</td>
<td>40</td>
<td>-</td>
<td>-</td>
<td>35</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>4</td>
<td>16</td>
<td>75</td>
<td>3</td>
<td>-</td>
<td>75</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>8</td>
<td>32</td>
<td>150</td>
<td>7</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>m4.4xlarge</td>
<td>16</td>
<td>64</td>
<td>300</td>
<td>13</td>
<td>-</td>
<td>300</td>
</tr>
<tr>
<td>m4.10xlarge</td>
<td>40</td>
<td>160</td>
<td>750</td>
<td>35</td>
<td>3</td>
<td>750</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>4</td>
<td>8</td>
<td>55</td>
<td>-</td>
<td>-</td>
<td>75</td>
</tr>
<tr>
<td>c4.2xlarge</td>
<td>8</td>
<td>15</td>
<td>135</td>
<td>3</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>c4.4xlarge</td>
<td>16</td>
<td>30</td>
<td>290</td>
<td>10</td>
<td>-</td>
<td>300</td>
</tr>
<tr>
<td>c4.8xlarge</td>
<td>36</td>
<td>60</td>
<td>590</td>
<td>25</td>
<td>-</td>
<td>650</td>
</tr>
<tr>
<td>r3.large</td>
<td>2</td>
<td>15</td>
<td>35</td>
<td>3</td>
<td>-</td>
<td>35</td>
</tr>
<tr>
<td>r3.xlarge</td>
<td>4</td>
<td>31</td>
<td>75</td>
<td>11</td>
<td>-</td>
<td>75</td>
</tr>
<tr>
<td>r3.2xlarge</td>
<td>8</td>
<td>61</td>
<td>155</td>
<td>26</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>r3.4xlarge</td>
<td>16</td>
<td>122</td>
<td>315</td>
<td>57</td>
<td>2</td>
<td>300</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>32</td>
<td>244</td>
<td>635</td>
<td>120</td>
<td>8</td>
<td>600</td>
</tr>
</tbody>
</table>

Assumptions:
• 2 GB RAM reserved for OS and base Spotfire processes.
• Numbers quoted are for a single analysis file.
• All users share 80% of data. Higher sharing allows more concurrent users and vice versa.
• The backend database can support the number of concurrent users indicated in the table.
SCALING SPOTFIRE FOR AWS
There are two primary ways to scale Spotfire for AWS:

- Scale up: Also known as scaling vertically is the process of making more system resources available to an existing Spotfire instance.
- Scale out: Also known as scaling horizontally is the process of adding new instances to an existing Spotfire deployment.

First, you need to determine when you need to scale your Spotfire deployment.

DETERMINING WHEN TO SCALE
Consider scaling your instance when any of the following occur:

<table>
<thead>
<tr>
<th>PERFORMANCE INDICATION</th>
<th>EXPECTED BEHAVIOR</th>
<th>POTENTIAL REMEDIATION</th>
<th>ADDITIONAL NOTES ON SCALING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow response to user interactions.</td>
<td>Response times should average 10-300 ms for most user interactions, for example, changing filtering or axes.</td>
<td>If response times are slower, you may need to:</td>
<td>You may scale up or out. Scaling up is fastest and easiest. Response times could be constrained by CPU, RAM, or I/O. Check your system performance metrics to identify bottlenecks and choose an EC2 instance size accordingly.</td>
</tr>
<tr>
<td>Slow data loading from information links</td>
<td>Loading data from information links should be comparable to querying the underlying database directly, with minimal overhead.</td>
<td>If data loads slowly from information links, you may need to:</td>
<td>• You may scale up or out. Generally, information links are not memory-constrained. If you find a CPU bottleneck, choose an EC2 instance size with more available CPUs.</td>
</tr>
</tbody>
</table>

- Response times are constrained by CPU, RAM, or I/O.
- Check your system performance metrics to identify bottlenecks and choose an EC2 instance size accordingly.
- Optimize your Spotfire analysis.
- Optimize your network connections (for in-database sources).
- Add a Spotfire instance in an EC2 region closer to users.
- Scale Spotfire to handle larger data sizes and/or load.
- Optimize the underlying database for query performance.
- Optimize the network to your database.
- Locate your Spotfire instance in an EC2 region closer to the database.
- Scale Spotfire to handle increased load.
<table>
<thead>
<tr>
<th>PERFORMANCE INDICATION</th>
<th>EXPECTED BEHAVIOR</th>
<th>POTENTIAL REMEDIATION</th>
<th>ADDITIONAL NOTES ON SCALING</th>
</tr>
</thead>
</table>
| Low on system memory (RAM) | For in-memory data, Spotfire will grow its memory consumption as data size grows. The amount of available RAM will limit the amount of data you can bring in-memory into Spotfire. | If your system is low on RAM, you may need to:  
• Scale Spotfire to handle larger data sizes and/or load.  
• Migrate your data to an analytics-oriented database, such as Amazon Redshift, and then use in-database access in Spotfire. | • You may scale up or out. Scaling up is easy and fast, and Amazon’s R3 instances are optimized for memory-intensive applications, such as Spotfire’s in-memory data engine. |
| Windows paging to virtual memory | As memory needs increase, Windows starts paging data from memory out to disk (virtual memory). Virtual memory paging by the operating system will adversely affect performance, as the O/S will block Spotfire’s in-memory data engine while repeatedly swapping data between RAM and disk. | Similar to the above, the best option is to scale the system so that Spotfire has access to more RAM. | You may scale up or out. Scaling up is easy and fast, and Amazon’s R3 instances are optimized for memory-intensive applications, such as Spotfire’s in-memory data engine. |

**MIGRATING FROM SQL SERVER EXPRESS**

The Spotfire AMI in the AWS Marketplace comes with Microsoft SQL Server Express preinstalled. This edition of SQL Server can access only a limited amount of system resources (for example, 10 GB storage, one processor, and 1 GB RAM). The Express edition will limit the number of analyses you can store with embedded data.

As you shift from evaluation and development to production, we recommend migrating the Spotfire database to the Standard or Enterprise edition of SQL Server. The steps are as follows:

• Move the database contents from SQL Express instance to production instance using MSSQL export/import.
• Update the Spotfire bootstrap file to point to the new database using the Spotfire command-line config tool.
HOW TO SCALE UP (OR DOWN)
AWS makes scaling up or down easy. With a variety of Amazon EC2 instance
types, you can select the most cost-effective instance type according to the
required system capacity and expected compute tasks. You can scale up or down
in minutes with these simple steps:
• Stop the Spotfire instance you want to re-scale.
• Select the desired instance type and size (larger or smaller).
• Restart the instance.
You may scale up until you reach the limits of the largest EC2 instance (for
example, r3.8xlarge).

HOW TO SCALE OUT
Scaling out is a more advanced setup, in which multiple EC2 instances containing
Spotfire services can be:
• Load-balanced - A cluster of identical services allows configuration for high
availability or for increased load.
• Service-optimized - A cluster of heterogeneous services allows each
component the exclusive use of the resources available on a single EC2
instance, optimized for the resource needs of the service.

LOAD BALANCED CONFIGURATION
Figure 2 shows a high availability setup where instances are behind a set of
reverse proxies.
This setup reduces the risk of failure on a single instance. You will also increase
your overall Spotfire capacity and possibly performance.

Figure 2 : High Availability Setup
It is also worth noting that you can simply deploy Spotfire for AWS in the middle tier if you are simply looking for high availability and not concerned with scaling each individual application separately.

If you choose to scale each application separately as illustrated in Figure 2, here are our recommendations:

<table>
<thead>
<tr>
<th>SPOTFIRE SERVICE</th>
<th>RECOMMENDED EC2 INSTANCE TYPE / MODEL</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spotfire Server</td>
<td>m3.large</td>
<td>The Spotfire Server requires minimal compute resources.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Save costs by using a general-purpose EC2 instance type.</td>
</tr>
<tr>
<td>Web Player</td>
<td>r3.large or larger</td>
<td>For in-memory analyses, the Web Player service requires enough RAM for the expected shared data size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spotfire is very memory efficient; as user sessions share exactly the same data and analyses, you will see only a small incremental increase in system resources for each additional user.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use a memory-optimized EC2 instance type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose an instance model (for example, large to 8xlarge) based on expected data size and user load.</td>
</tr>
<tr>
<td>Automation Services</td>
<td>r3.<em>, m3.</em> or t2.*</td>
<td>Automation Services is very similar to the Web Player, but without a web UI for direct end-user interaction. As such, it will have very similar system resource demands as the Web Player (memory-bound), but without the session scaling demands.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each automation job runs in a separate process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automation Services is a good candidate for bursting, launching instances only as automation tasks are scheduled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Otherwise, Automation Services can run in parallel on the Web Player instance, using compute capacity at night while the Web Player load is low.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use a memory-optimized, general-purpose or burstable instance type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose an instance model (for example, large to 8xlarge) based on expected data size and number of parallel automation jobs (generally one per processor).</td>
</tr>
</tbody>
</table>
You can isolate these services and ensure they have dedicated system resources with a few simple steps for each service type:

**CONFIGURING A SEPARATE SPOTFIRE SERVER INSTANCE**
- Stop all Spotfire services on the Spotfire instance except for the Spotfire Server
- (OPTIONAL) Load-balance the Spotfire Server
  - Start a new instance of Spotfire in the AWS Marketplace
  - Stop all Spotfire services on the new instance except for the Spotfire Server

**CONFIGURING A SEPARATE WEB PLAYER INSTANCE**
- Start a new instance of Spotfire in the AWS Marketplace
- Stop all Spotfire services on the new instance except for the Web Player
- Change the new instance’s Web Player configuration to point to the existing Spotfire Server instance (using the existing instance’s hostname instead of localhost)

```xml
<spotfire.dxp.web>
  <setup>
    <authentication serverURL=http://localhost:8080 ...>
    • Restart the Web Player service on the new instance.
    • (OPTIONAL) Stop the Web Player service on the original Spotfire instance.
  </setup>
</spotfire.dxp.web>
```

**CONFIGURING A SEPARATE AUTOMATION SERVICES INSTANCE**
- Start a new instance of Spotfire in the AWS Marketplace
- Stop all Spotfire services on the new instance except for Automation Services
- Change the new instance’s Automation Services configuration to point to the existing Spotfire Server instance (using the existing instance’s hostname instead of localhost). This is done by editing the “serverurl” property in the following file:
  
  ```
  C:\Program Files\TIBCO\Automation Services\webroot\bin\Spotfire.Dxp.Automation.Launcher.exe.config
  ```
- (OPTIONAL) Stop any Automation Services related schedules on the original Spotfire instance.

**LEARN MORE**
For more detailed instructions on how to configure Spotfire for advanced deployments, see the following documents:
- Spotfire Server Installation and Configuration Manual
- Spotfire Web Player Installation and Configuration Manual
- Spotfire Automation Services Installation and Deployment Manual