

如何写入更有效的消息过滤器？

目录

[问题](#)

问题

如何写入更有效的消息过滤器？

当消息过滤器变得更加长，他们能影响您的ESA性能特性。对于很小数量的过滤器或短过滤器，效率不是一重大的注意事项。然而，当修建更加长的过滤器或，如果您的实施有许多过滤器时，您应该是记住某些操作相对效率。

当通过消息通过消息渠道时，所有消息过滤器被结合到被评估用每个消息的一个基本方式的单个表达式。这意味着过滤器排序是非常重要的，并且能短路复合表达式的进一步评估。例如，如果有将适用于消息的一定数量的过滤器，但是一个过滤器非常频繁地将应用并且有最后的行动deliver()、bounce()或者drop()关联与它，一样及早在列表应该移动过滤器尽可能。

虽然ESA是非常高效在其处理常规表达，您能滥用常规表示引擎就象导致另外或多余处理。常规表示的每个评估大致采取相同数量资源，因此含义那减少您评估表达式的数量将产生更加了不起的效率。例如，在以下过滤器，在每个“丢弃附件由NAME的”常规表达是单个被评估的全部，含义常规表示评估发生7次，当比较附件名称对在丢弃附件由NAME时的模式：

```
strip_all_dangerous () {  
NAME `(i) \ \ .pif$`);  
NAME `(i) \ \ .exe$`);  
NAME `(i) \ \ .scr$`);  
NAME `(i) \ \ .msi$`);  
NAME `(i) \ \ .java$`);  
NAME `(i) \ \ .dll$`);  
NAME `(i) \ \ .com$`);  
}
```

在以下示例中，结果是等同的，但是示例是更有效的，导致仅单个常规表示评估：

```
strip_all_dangerous () {  
NAME `(i) \ \ (pif|exe|scr|msi|Java|dll|com) $`);  
}
```

虽然第二常规表示比七部分复杂在第一个过滤器，它比七简单一个是评估一复杂常规表示的更有效的。

然而，此技术一定是平衡的开销维护这样过滤器。