

户端连接 Agent 的数量。在这种情况下不建议设置 threads 这个参数。如果使用更新版本的 Thrift, Flume 使用 Thrift 的 TThreadedSelectorServer, 使用 Java 非阻塞式 I/O, 因此可以支持比可用的线程更多的客户端。在这种情况下, threads 参数和 Avro Source 的 threads 参数工作类似, 可以用来控制资源利用率。

Thrift Source 不同于 Avro Source, 目前不支持压缩或 SSL。因此 Thrift Source 只能用来从非 JVM 语言系统推送数据到 Flume 中, 或者从用于别的用途的 Thrift 的写数据的应用推送数据到 Flume 中。对于 Flume Agent 到 Flume Agent 的通信, 推荐使用 Avro Sink-Avro Source 对。

下面是 Thrift Source 配置的例子 :

```
agent.sources = thriftSrc
agent.channels = memChannel

agent.sources.thriftSrc.type = thrift
agent.sources.thriftSrc.channels = memChannel

# Bind to all interfaces
// 绑定到所有的接口
agent.sources.thriftSrc.bind = 0.0.0.0
agent.sources.thriftSrc.port = 4564

# Initializes a memory channel with default configuration
// 使用默认配置初始化 memory channel
agent.channels.memChannel.type = memory
```

写入这个 Source 的 Thrift Sink 将会有如下类似的配置 :

```
agent.channels = memChannel
agent.sinks = thriftSink

agent.channels.memChannel.type = memory

agent.sinks.thriftSink.type = thrift
agent.sinks.thriftSink.channel = memory
agent.sinks.thriftSink.hostname = thriftsrc.example.com
agent.sinks.thriftSink.port = 4564
```

RPC Sources 的失败处理

在 Avro Source 和 Thrift Source 中失败的处理都有些棘手。这是因为 RPC Source 尽管看起来像是本地方法的调用, 但实际是被另一边网络链路上的客户端或 Sink 调用。由于参