"Why JMeter and New Relic Report Different Response Times: A Deep Dive into Client vs. Server-Side Metrics"

The discrepancy between **JMeter's response times** (750 ms at 90th percentile) and **New Relic APM's response times** (150 ms at 90th percentile) typically occurs due to differences in how each tool measures and reports response time.

Here's a detailed, technical explanation of why these differences arise:

---

① Measurement Perspective

- **JMeter (Client-Side Measurement)**
  Measures **End-to-End response time**, including:
  - Network latency (client → server → client).
  - DNS lookup time, TCP connection time, SSL handshake.
  - Server processing time.
  - Data transfer times (download of response payload).
- **New Relic APM (Server-Side Measurement)**
  Captures only **server-side processing time** (pure execution of the application logic):
  - Does **NOT** include network latency from client to server.
  - Does **NOT** measure client-side rendering or network overhead.
  - Reports only the duration of code execution on the backend.

---

② Key Contributors to the Difference

Factor	Included in JMeter?	Included in New Relic?
DNS Resolution Time	✅ Yes	❌ No
TCP Connection Establishment	✅ Yes	❌ No
TLS/SSL Handshake	✅ Yes	❌ No
Request Transfer Time (Upload)	✅ Yes	❌ No

Server Processing Time (Pure App)	✅ Yes	✅ Yes
Response Transfer Time (Download)	✅ Yes	❌ No
Client-side Latency	✅ Yes	❌ No

---

③ Technical Scenarios & Examples

- **Scenario 1: Network Latency**
  If the server is geographically distant from the JMeter client machine, network latency can significantly increase total response time observed by JMeter, while New Relic remains unaffected as it reports only server execution time.

Example:

- ○ **JMeter**: Server execution (150 ms) + Network latency (500 ms) + Payload transfer (100 ms) = **750 ms**.
- ○ **New Relic**: Server execution alone = **150 ms**.

- **Scenario 2: Payload Size**
  Large response payload sizes increase download time, inflating JMeter response times. New Relic doesn't reflect this overhead.

---

④ Practical Analysis Steps

If you want to precisely pinpoint the source of this discrepancy, try these steps:

1. **Network Analysis**
   - ○ Use a network profiling tool (Wireshark or Fiddler) to measure latency and bandwidth issues separately.
   - ○ Observe DNS, TCP, SSL handshake times distinctly.
2. **JMeter Timers (Transaction Breakdown)**
   - ○ Use JMeter's built-in timers and listeners (e.g., View Results Tree with detailed latency and connect time metrics).
3. **New Relic Transaction Breakdown**

    - Use Transaction Traces in New Relic to confirm the exact server processing time and verify against database, third-party API, or internal processing delays.

4. **Browser-side Validation (optional)**
    - Validate with real browsers (DevTools Network tab) to cross-reference response times measured on client-side.

---

⑤ Recommended Resolution or Clarifications

- **Acceptable Difference**:
  It's normal to see a 2-4x difference between APM and JMeter due to network overhead.
- **Reduce Latency or Distance**:
  Use load generators closer to the server to minimize network impact.
- **Segment Reporting**:
  Clearly distinguish reports between **server-side performance** (New Relic) and **End-to-End user experience** (JMeter).
- **Explain in Documentation**:
  Document clearly that JMeter times are "client perceived" and include additional network overhead, whereas New Relic strictly reports application response times.

📌 Conclusion & Next Steps

The difference you're observing is expected and primarily attributed to **network latency and client-side overhead** included by JMeter but excluded from New Relic's pure server-side measurement.

Suggested Follow-up:

- Run a localized load test (JMeter on same network/data center as the server) to validate if the response times converge.
- Analyze New Relic breakdown further (e.g., database vs app logic breakdown) to rule out internal server delays.

This approach ensures clarity on your performance metrics and appropriate reporting to stakeholders.