



ロードバランスを考える

株式会社サイバーエージェント
田中 淳



- 田中 淳
- CAのインフラネットワークを担当
- L7～L1までやってます



コミュニティ



ゲーム





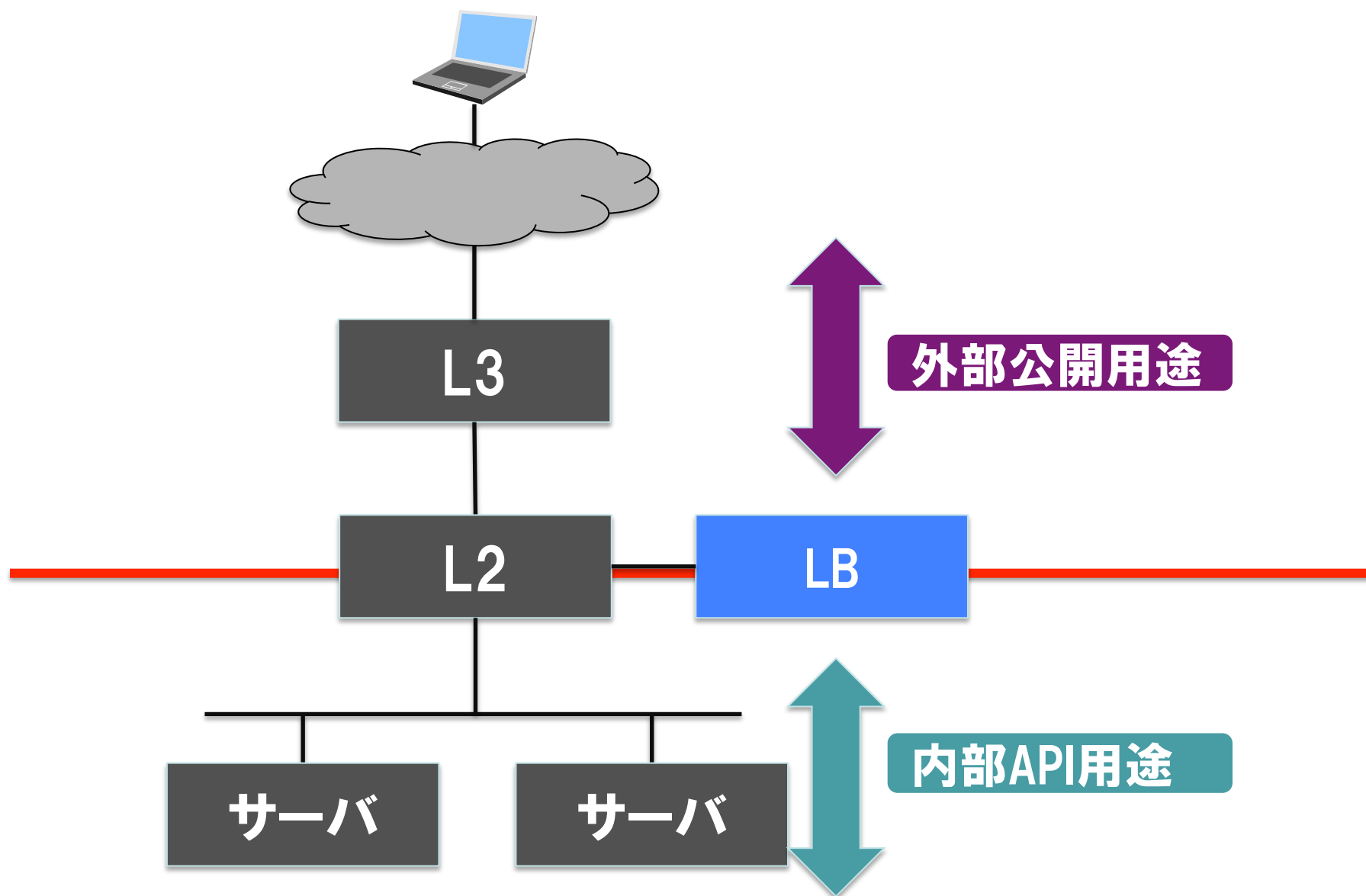
ベンダー機とLVSの割合

3 : 1

ベンダー

LVS

※全DCで約40台いました(汗)





■ 内部API用途での方法

- L2DSR
- L4 + SNAT (L7 + SNAT)

■ 外部公開用途での方法

- L4
- L7 (SSLなど)



■ 内部API用途での方法

- L2DSR
- L4 + SNAT (L7 + SNAT)

■ 外部公開用途での方法

- L4
- L7 (SSLなど)



■内部API用途

1. リクエスト

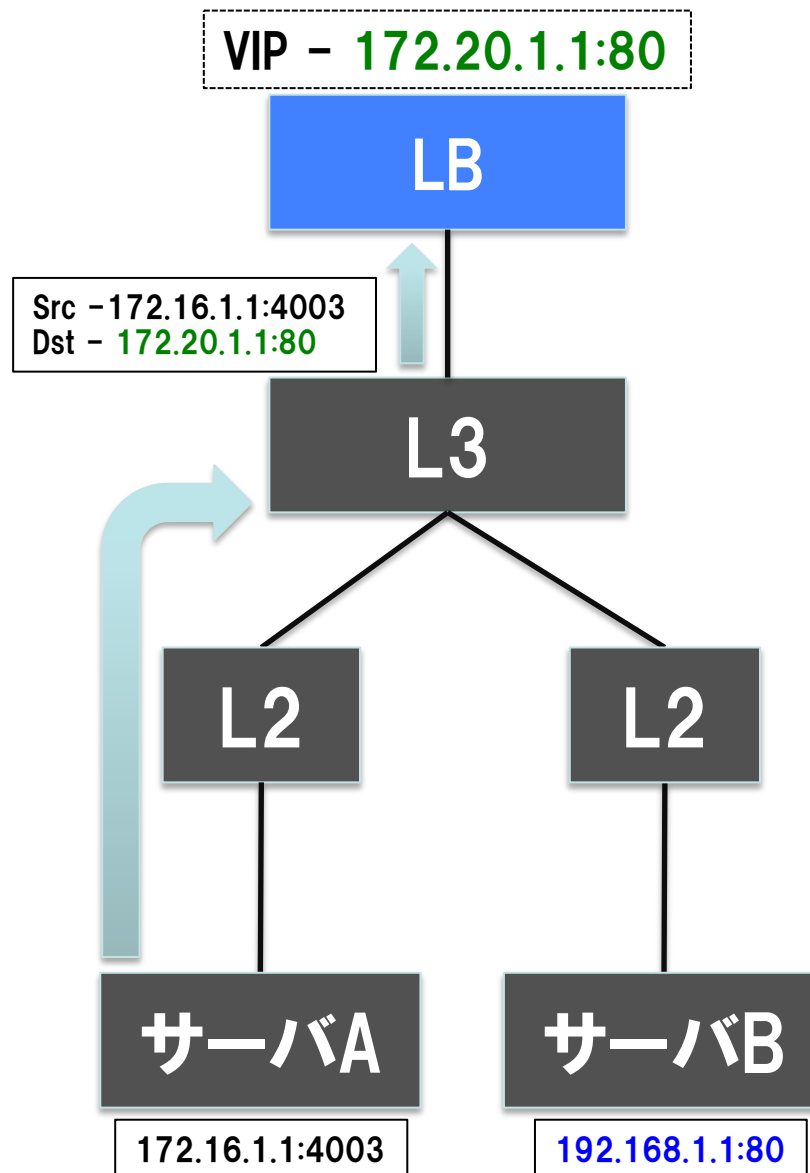
1. サーバA→VIP80番リクエスト

2. SNAT変換

1. LBでSrcアドレス変換 (SNAT)
2. サーバBで受けるSrcがSNAT IP

3. レスポンス

1. サーバBはSNAT IPへ応答
2. LB→サーバAへ (SrcはVIPに戻る)





■内部API用途

1. リクエスト

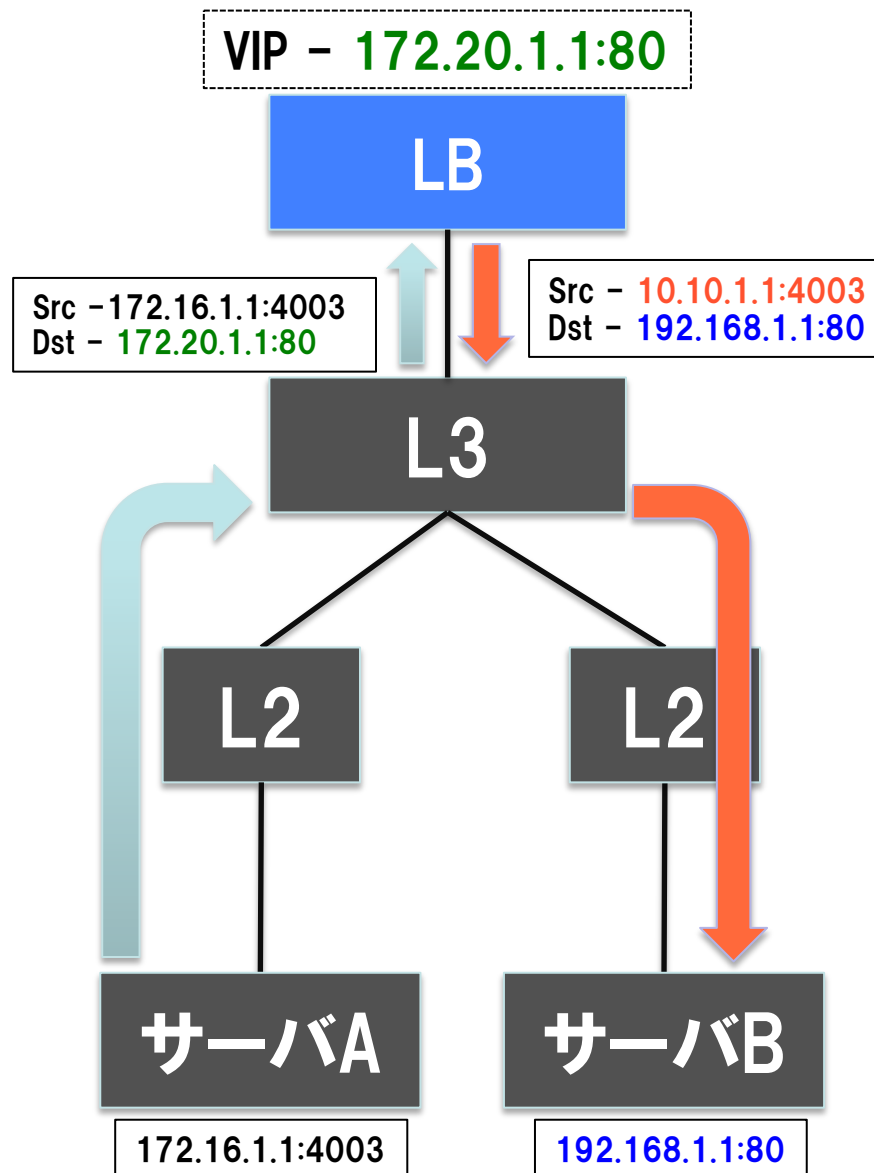
1. サーバA→VIP80番リクエスト

2. SNAT変換

1. LBでSrcアドレス変換 (SNAT)
2. サーバBで受けるSrcがSNAT IP

3. レスポンス

1. サーバBはSNAT IPへ応答
2. LB→サーバAへ (SrcはVIPに戻る)





■内部API用途

1. リクエスト

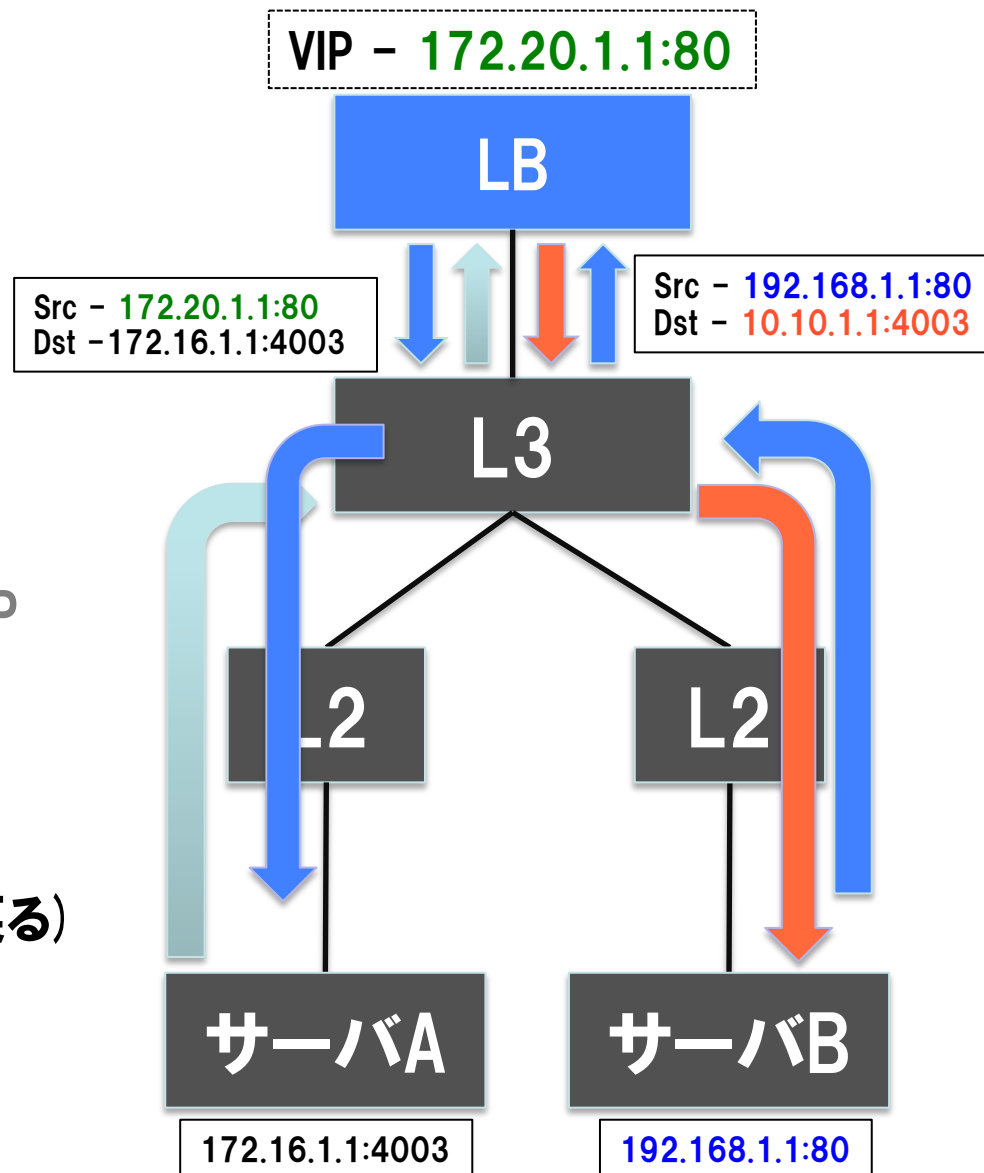
1. サーバA→VIP80番リクエスト

2. SNAT変換

1. LBでSrcアドレス変換 (SNAT)
2. サーバBで受けるSrcがSNAT IP

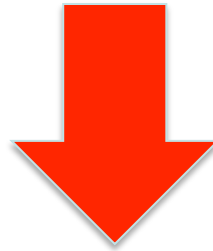
3. レスポンス

1. サーバBはSNAT IPへ応答
2. LB→サーバAへ (SrcはVIPに戻る)





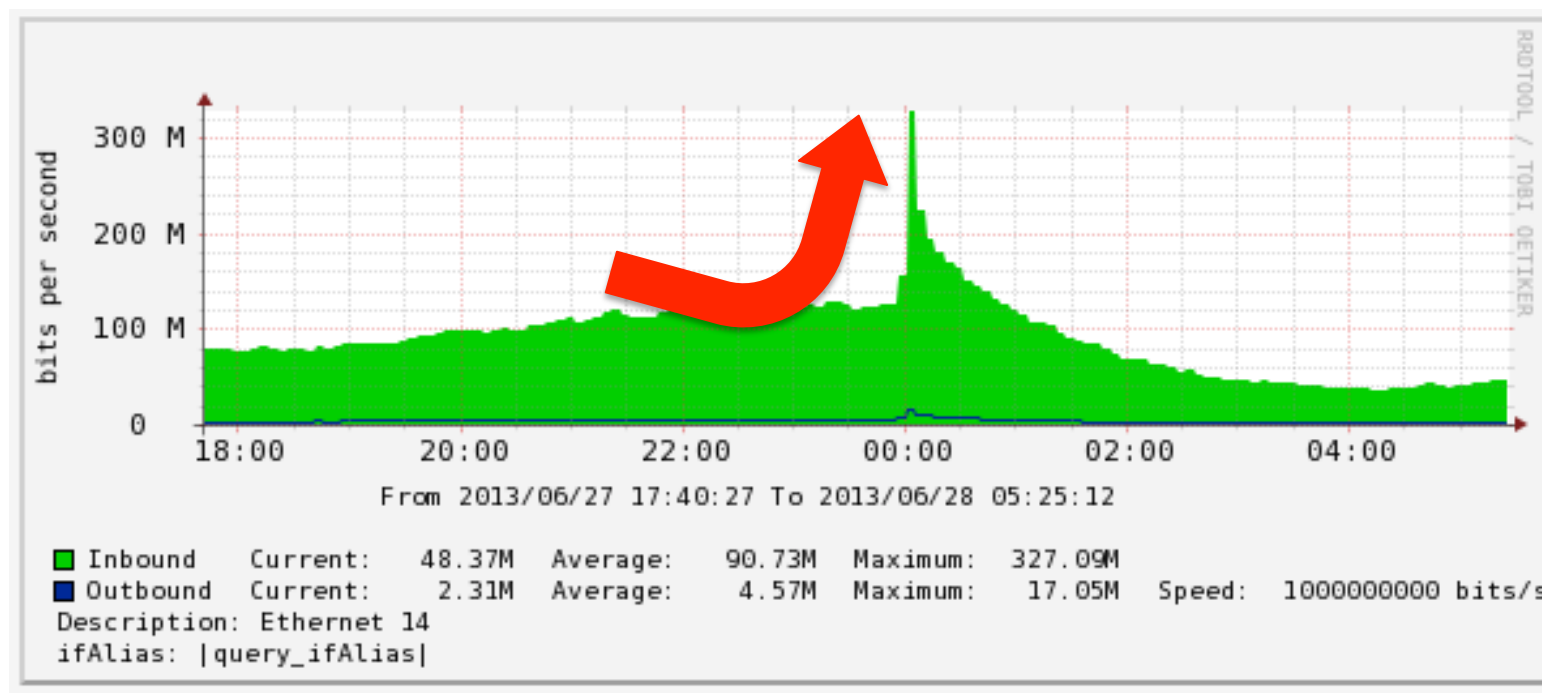
よく起こる問題



SNAT Port reused問題



このタイミング





■ 現象について

1. サービスが盛り上がる
2. SNAT変換急増 ↗
3. サーバがもつTIMEWAIT情報＝LBの
SNAT IP&ポートかぶる・・・



■内部API用途

1. レスpons

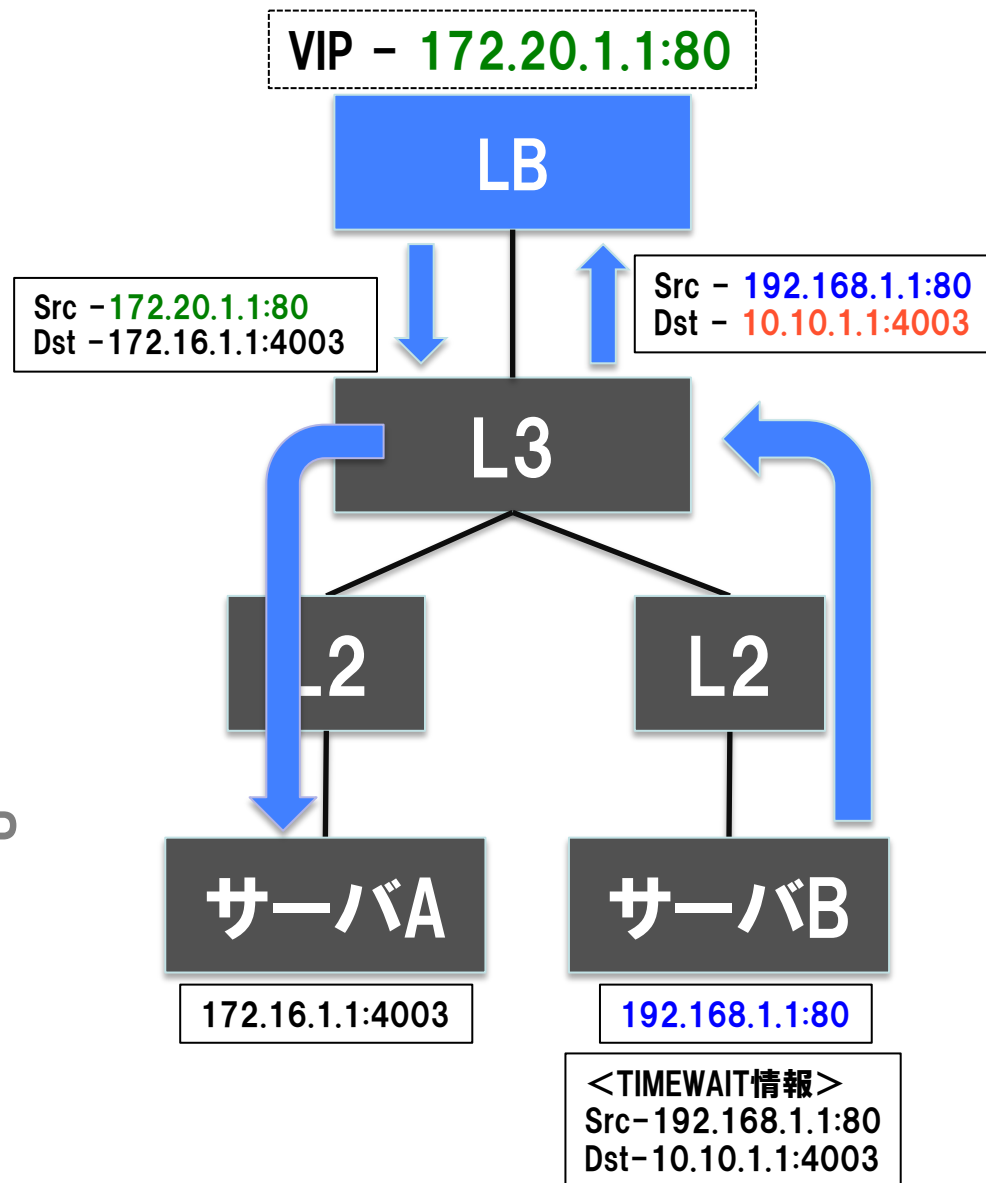
1. サーバBはSNAT IPへ応答
2. LB→サーバAへ

2. リクエスト

1. サーバC→VIP80番リクエスト

3. SNAT変換

1. LBでSrcアドレス変換 (SNAT)
2. サーバBで受けるSrcがSNAT IP





■内部API用途

1. レスponse

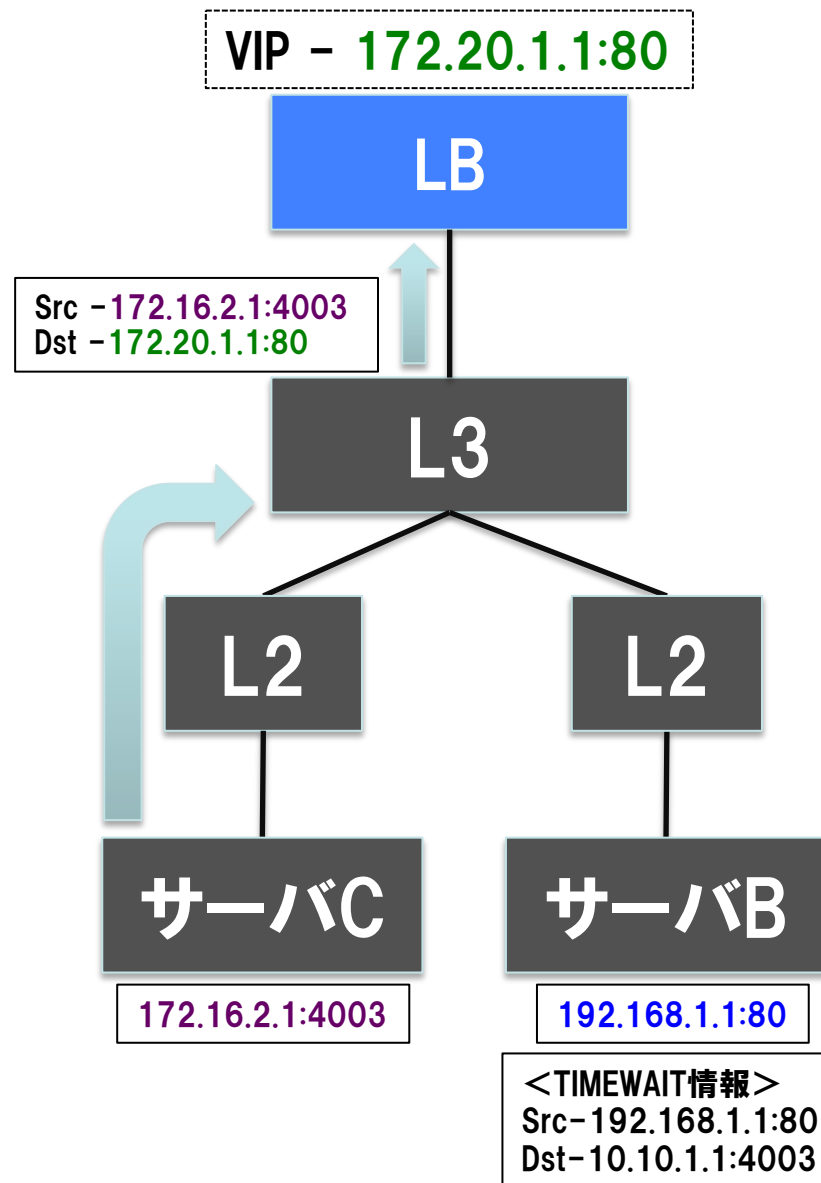
1. サーバBはSNAT IPへ応答
2. LB→サーバAへ

2. リクエスト

1. サーバC→VIP80番リクエスト

3. SNAT変換

1. LBでSrcアドレス変換 (SNAT)
2. サーバBで受けるSrcがSNAT IP





■内部API用途

1. レスpons

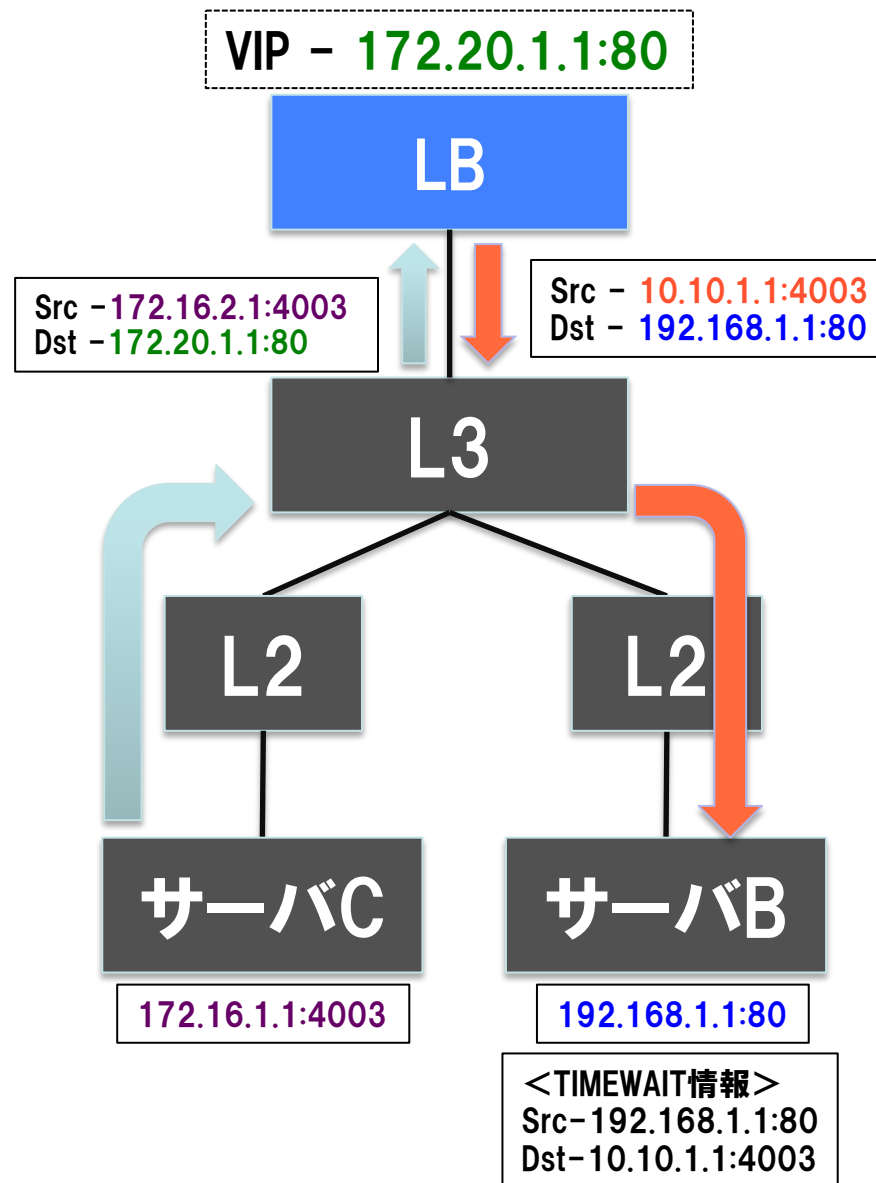
1. サーバBはSNAT IPへ応答
2. LB→サーバAへ

2. リクエスト

1. サーバC→VIP80番リクエスト

3. SNAT変換

1. LBでSrcアドレス変換 (SNAT)
2. サーバBで受けるSrcがSNAT IP





■内部API用途

1. レスponse

1. サーバBはSNAT IPへ応答
2. LB→サーバAへ

2. リクエスト

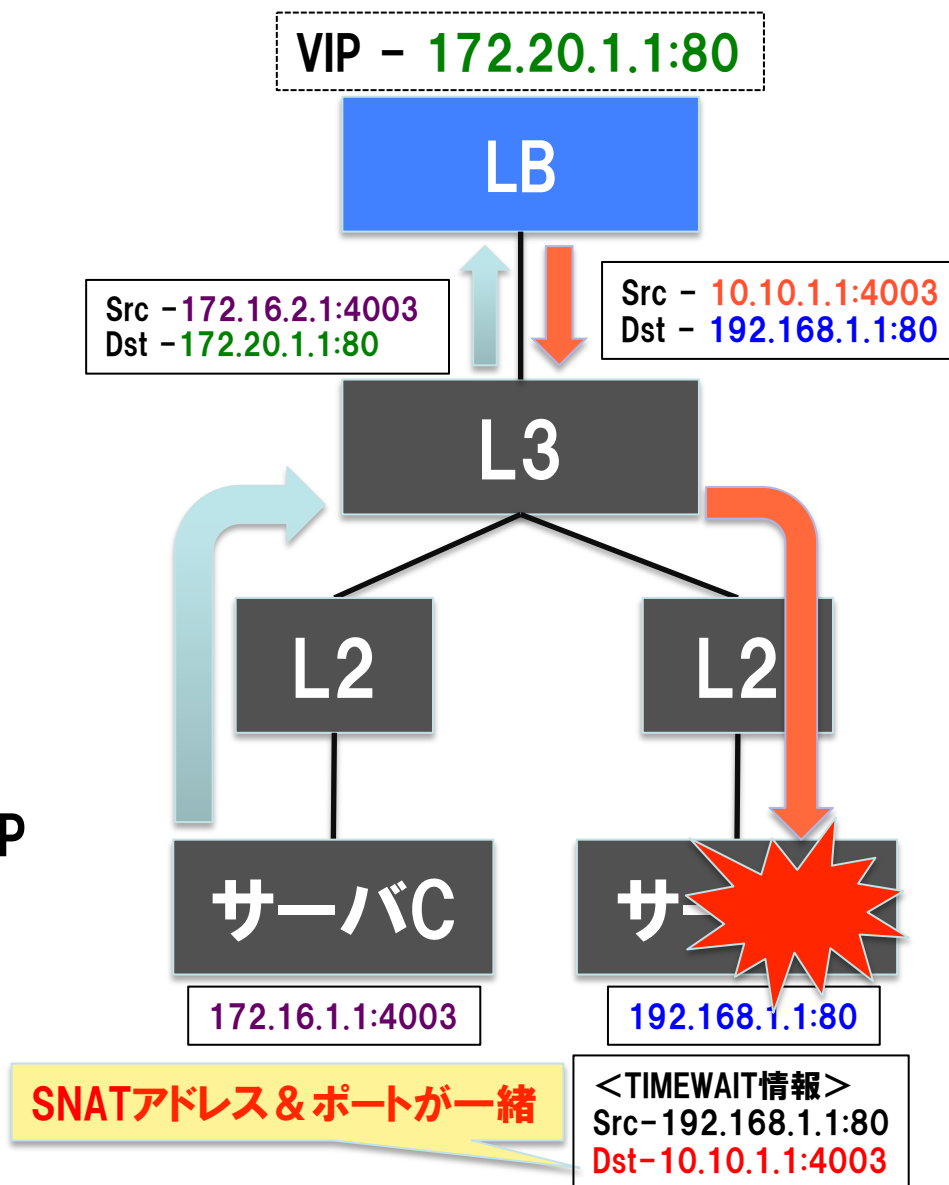
1. サーバC→VIP80番リクエスト

3. SNAT変換

1. LBでSrcアドレス変換 (SNAT)
2. サーバBで受けるSrcがSNAT IP

4. レスponse (Error)

1. Srcアドレス&ポートが一緒！





■ 対策

1. SNATアドレスを増やす
2. サーバのTIMEWAIT値をめちゃくちゃ短くする
3. LB側でテンプレートを適用 (TIMEWAIT値を変更)



■ 悩めること

1. SNATアドレスを増やす

- アドレス増やしてもSNAT変換のハッシュでまた同じの使っちゃう可能性有り(確率論的な話)

2. サーバのTIMEWAIT値をめちゃくちゃ短くする

- 他に影響出ちゃいそうだったのでやりませんでした。

3. LB側でテンプレートを適用(TIMEWAIT値を変更)

- L7で処理させる必要あり。負荷が・・・



他の悩み

1. 分散、集約って繰り返すけど、集約のほうがいいのかな？
2. 接続する機器同士の分散アルゴリズムがいけてないと、LAG組んだ時が泣ける・・・。
3. 高機能な分色々任せると、これまた泣ける・・・。
4. ベンダーが分散していると運用する人が覚えるのが大変。
5. やっぱり、アプライアンス高いっす (T T)

などなど