왕자영요(王者荣耀) 백그라운드 개선 Proxy편

* [yongzhang](https://km.woa.com/user/yongzhang)

Tencent 게임 클라우드 솔루션 아키텍트 Nelson:

왕자영요는 세계에서 가장 인기 있는 모바일 게임 중 하나이며 DAU의 수익이 1억을 돌파했습니다. 이렇게 많은 수의 사용자를 수용하는 방법은 왕자영요의 백엔드 아키텍처에 대한 도전 과제입니다. 그리고 왕자영요의 플레이 방식은 PVP이기 때문에 서버의 지역 구분 없이 다른 지역의 플레이어, 다른 플랫폼의 플레이어, iOS 및 Android 플레이어들이 매칭되어 PVP로 게임을 진행할 수 있습니다. 백그라운드는 로비 서버와 PVP 서버로 구분되고 PVP 서버의 사용은 CGI 호출과 유사하며 리소스를 배정받아 사용 후 재활용할 수 있습니다. 필요한 것은 로비에서 받고 사용 후 로비로 다시 보냅니다. DB의 라이트백(write-back)은 로비에서 진행됩니다. 따라서 로비와 PVP 서버 사이에 중간 포워딩이 필요합니다. 왕자영요에서는 이를 Proxy라고 부르는데 프록시 서버와 동등하며 자체 백엔드에서 많은 프로세스 배포의 세부 사항을 차폐합니다.

이것의 장점은 로비 서버의 뒤에 몇 개의 방 서버가 있는지 알 필요가 없고 룸 서버가 있다는 것만 알면 액세스할 수 있다는 것입니다. 부하를 어떻게 구분하고 밸런스를 유지하는지, 백엔드에서 고장 노드를 차폐하는 방법은 모두 Proxy 라우팅 기능을 통해 실현됩니다.

그러나 사용자가 증가함에 따라 매칭 규칙이 점점 더 복잡해지고 전체 라우팅을 담당하는 Proxy가 점점 더 복잡해집니다. Proxy 자체의 안정성과 성능은 왕자영요 백스테이지에서 매우 중요한 부분이 되었습니다. 이 기사에서는 왕자영요의 급속한 비즈니스 성장 과정에서 직면한 문제와 개선 아이디어를 공유합니다.

IEG/Tianmei L1 Studio/Server Team Staff

**왕자영요 Backstage Architecture**

**ダイアグラム

自動的に生成された説明**

Apollo 관련 및 공동 구성 요소

계정 인증, ID매핑

다수의 랩톱

기타

로직 파티션1~N

Gamesvr-roomd 그룹

PvoRoomMatch 그룹

Pvorelaysvr 그룹

Guildsvr그룹

Worldlogicsvr그룹

Proxy 클러스터

          왕자영요 백그라운드의 특징:

1. 운영 전략 측면으로 볼 때 지역과 서버가 구분되는 게임이지만 구현 측면으로 보면 전체 지역과 서버 아키텍처이며 버디와 클랜 등 시스템은 논리적 파티션 전반에 걸쳐 서비스를 제공합니다. gamesvr, roommatch 및 relaysvr은 모두 리소스 풀 형태로 서비스를 제공합니다.
2. 관계망 유형에 따라 데이터 저장 시스템의 구분 없이 매칭을 달성했습니다(예: 모바일 QQ Android 사용자는 IOS 장치로 게임을 하는 친구와 함께 플레이 가능).
3. 서비스 모듈은 그룹 형태로 존재하며 온라인에서 원활하게 확장할 수 있고 동시에 프로세스 간의 하트비트 유지 관리, 고장 노드 자동 차폐도 할 수 있습니다.

DAU 및 PCU의 지속적인 증가에 따라 확장 진행중입니다. PCU 확장 시 일반적으로 gamesvr(로비 관련 모든 논리적 담당)와 relaysvr(프레임 동기화 전투 서비스) 기기를 비례하여 증가시키면 됩니다. 새로운 논리적 파티션을 추가할 때도 worldlogicsvr(논리적 파티션 등록 제한 등 관련 논리적 등 담당) 및 guildsvr(클랜 서비스)를 증가시키면 됩니다. 오랫동안 확장 과정은 순조롭게 진행되었지만 결국 문제가 발생했습니다.

**Proxy CPU 문제 개선**

**배경:**

올해 5월 중순 금요일 오후 5시부터 모바일 QQ Android 영역의 proxy 머신은 CPU가 90+%-100%에 도달하는 문제가 발생했습니다.

**긴급 조치:**

이 영역에서 proxy 확장 수행

**긴급 처리 효과:**

초당 proxy 프로세스가 처리하는 데이터 패킷의 양이 원래 3w+/s에서 2w8/s로 감소했지만 CPU는 뚜렷하게 낮아지지 않고 여전히 90% 이상입니다.

**문제 분석:**

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

Perf는 proxy의 주요 소비가 tbus\_peek\_msg이고 apollo 동료들과 검토한 결과 tbus의 메시지 읽기는 채널별 폴링 방식을 기반으로 합니다. 왕자영요 모바일 QQ의 Android 관련 기기는 850개 이상이 있습니다. proxy는 SS 메시지 허브로서 다른 모든 유형의 프로세스와 통신하는 통로를 만들어야 하는데 피크 기간 동안 엄청난 양의 메시지와 많은 사이클 수로 인해 CPU가 급증했습니다.

**Proxy 개선 솔루션:**

문제는 메시지를 읽을 때 소비한 시간입니다. Proxy는 배정에 따라 시동 시 읽기만 담당하는 자식 프로세스를 만듭니다. 메인 프로세스에서는 포워딩만 쓰고 메시지를 읽는 프로세스는 tbus의 채널을 균등분배 후 배정받은 채널만 담당합니다. 읽기 프로세스와 쓰기 프로세스 사이에 shm을 기반으로 하는 락 프리 큐(Lock Free Queue)를 통해 통신합니다.

ダイアグラム

自動的に生成された説明

Proxy 프로세스 쓰기

Proxy 프로세스 읽기

Proxy 프로세스 읽기

**Proxy 개선 효과:**

업데이트된 cpu 스크린샷은 다음과 같습니다.

文字と写真のスクリーンショット

自動的に生成された説明

현재 proxy를 담당하는 기계는 6코어 기계로 4개의 읽기 프로세스를 할당되어 있습니다. 4개의 읽기 프로세스 cpu가 약 10%이고 쓰기 프로세스는 약 5%입니다.

Proxy CPU 문제 해결되었지만 새로운 문제가 다시 발생했습니다...

**Proxy 메모리 개선**

**Proxy 새로운 문제:**

Proxy 머신이 더 이상 메모리를 늘릴 수 없고 Android 모바일 QQ 영역이 확장할 수 없다는 O&M 피드백

**이유:**

Proxy는 다른 모든 프로세스와의 통신 채널을 만들어야 합니다. gamesvr 및 relaysvr를 확장하는 것은 tbus 메모리를 계속 늘리는 것을 의미합니다. Proxy는 40G에 가깝습니다.

**일단 막아내고 나중에 개선:**

현재 네트워크 tbus 채널 용량을 모니터링한 후, tbus 채널 크기를 20M에서 10M으로 줄이고 계속 확장할 수 있도록 해야 합니다.

**Proxy 아키텍처 조정**

ジャック, 電子機器 が含まれている画像

自動的に生成された説明

Proxy 그룹2

Proxy 그룹1

Proxy 클러스터

Proxy 클러스터

Proxy의 현재 병목 현상은 메모리에 있습니다. 해결책은 Proxy를 그룹화하는 것입니다. 각 Proxy 그룹은 일부 gamesvr, relaysvr 및 기타 모든 프로세스를 담당합니다. Proxy는 서로 통신합니다. 목적지 주소가 이 그룹에 없으면 해당 Proxy 그룹으로 전달됩니다. 현재 왕자영요의 Proxy 서버 4대로 110대의 gamesvr와 250대의 relaysvr의 운전을 지원해 주고 있는데 Proxy 서버 한 대당 32G 용량을 차지하고 있습니다. 확장 시 그룹 서버 수가 가득 차지 않으면 그룹 내에서 확장을 하거나 그룹을 추가해야 합니다. 이로써 메모리 부족 문제를 해결할 수 있습니다.

**Proxy 개선 요약**

1. 이번 proxy 개선 과정은 대규모 서비스 운영 방침 중 막아내고 개선한다는 전략을 충분히 보여 주었습니다.
2. 서버 성능 모니터링 구축이 중요하다는 것을 보여주었습니다. 되도록 문제를 신속하게 발견한 후 막아내고 개선하는 시간을 벌어 줘야 합니다.