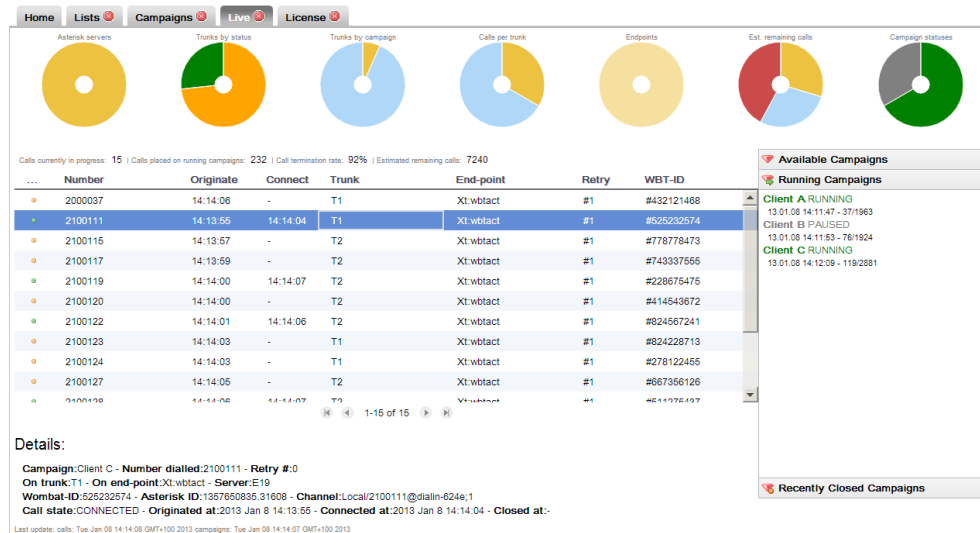


Lo mejor de la marcación automática



Tu WombatDialer es una herramienta muy potente: es capaz de adaptarse significativamente para gestionar miles de llamadas a la vez. Tanto si tienes 100 como 10.000 canales, no querrás chocar con un muro el día que pongas en marcha la producción.



En primer lugar, definamos lo que entendemos por marcación de alto volumen.

Para el propósito de este artículo, definimos "alto volumen" como una cantidad de llamadas que saturará un sistema Asterisk de stock.

Es difícil poner un valor específico aquí - depende del tipo de hardware que utilices y de lo que tu PBX esté haciendo realmente al procesar tus llamadas. Si realmente quieres poner un número aquí, digamos que podría ser alrededor de 250 llamadas, más o menos 150.

Optimizando Asterisk

Como Wombat se sustenta en Asterisk para la marcación, es importante entender cuáles son los desafíos para el procesamiento de grandes volúmenes de llamadas en su sistema Asterisk:

En primer lugar, lo que haces realmente importa. El número de llamadas que tu PBX gestionará con éxito depende de lo que estés haciendo en ella y de lo cara que sea cada llamada. Aquí hay tres sospechosos habituales: la decodificación, la grabación de llamadas y los scripts AGI.

La decodificación utilizará una gran cantidad de CPU para traducir el audio de un formato a otro. Esto ocurre silenciosamente cada vez que reproduces una grabación en un formato y tu canal utiliza un formato diferente. Tu música en espera es un MP3 y tus canales hablan G729 - eso es decodificar. Mientras que esta es una característica muy conveniente que Asterisk ofrece, es un no-no para la marcación de alto volumen.

La grabación de llamadas puede resultar muy costosa rápidamente, ya que utilizará valiosos IOPS de disco para escribir cientos de archivos de audio en el disco y valiosos ciclos de CPU para convertir el audio en su formato de almacenamiento favorito.

Si realmente necesitas la grabación de llamadas, debes asegurarte de que se active sólo cuando sea necesario (por ejemplo, cuando un cliente esté conversando con un agente), que tus discos sean tan rápidos como puedas conseguirlos (y un disco solido es una buena opción en este caso) y que el formato de almacenamiento requiera la menor cantidad de CPU posible.

Para la grabación a gran escala, el uso de una grabadora pasiva independiente (en red) puede ser una buena opción.

Los scripts AGI requieren que se ejecute un proceso en cada convocatoria. Se utilizan a menudo en Asterisk ya que proporcionan una forma sencilla de "pegar" cierta lógica y acceso a servicios externos (servicios web o bases de datos) al plan de marcación de Asterisk.

Suelen estar escritos en lenguajes de scripting lentos e interpretados y definitivamente no permitirán escalar. Cuando se ejecuta Wombat, suele ser una mejor opción obtener previamente cualquier valor de interés procedente de sistemas externos y hacer que Wombat lo pase a Asterisk como una variable de canal.

Por ejemplo, en lugar de ejecutar un script AGI que lanza un intérprete que abre una conexión de base de datos sobre la que se ejecuta una consulta SQL para decidir si se debe reproducir un mensaje en inglés o en español, esto podría ser una variable de canal establecida por Wombat que ya contiene el nombre de un archivo de audio personalizado para reproducir.

El segundo aspecto importante que debes tener en cuenta es si un sistema basado en una GUI es adecuado para ti. Aunque tener una GUI te facilita la vida a la hora de administrar el sistema, el tipo de plan de marcación que se genera suele ser extremadamente complejo y, por tanto, lento. Mientras que esto se hace con el fin de ofrecerte una tonelada de características interesantes para la PBX de tu oficina, puedes encontrar que realmente no las necesitas todas cuando haces marcación saliente a gran escala. Lo mejor de ambos mundos es utilizar un plan de marcación codificado a mano para marcar (la parte que es llamada por Wombat), y utilizar tu GUI favorito para todas las demás funciones.

Una tercer área de preocupación es causada por la interfaz AMI que Wombat utiliza para manejar Asterisk. El protocolo AMI puede volverse extremadamente "charlatán", ya que el sistema genera muchos eventos, y esto puede llevar a una latencia significativa antes de que todos los eventos sean procesados.

Esto puede solucionarse filtrando el conjunto de eventos que Asterisk envía (por ejemplo: Wombat no necesita eventos de ejecución de planes de marcado), pero incluso en este caso en algún momento el problema se manifestará. Además, Asterisk sólo puede procesar un número determinado de peticiones por segundo, y en algún momento querrá generar demasiadas llamadas para que tu sistema Asterisk pueda manejarlas.

Un último y general inconveniente con Asterisk es que procesa los canales de voz; si el sistema está sobrecargado y la voz no se procesa en tiempo real, te encontrarás con problemas de calidad de voz.

Llamar a la gente y ofrecerles un mensaje de audio roto o ininteligible es un gran no-no. Por ello, debes asegurarte de que tienes suficientes "buffers de capacidad" en tus sistemas para que la calidad de la voz se mantenga nítida en cualquier circunstancia.

Por lo tanto, la verdadera solución para la marcación de grandes volúmenes es el escalamiento. Esto significa ejecutar múltiples servidores Asterisk en paralelo, cada uno de los cuales está procesando sólo un conjunto limitado de canales que está bien dentro de sus límites. Por suerte, Wombat fue diseñado desde el principio para manejar un conjunto de servidores Asterisk como si fueran una sola máquina más grande. Para Wombat casi no hay diferencia en procesar 500 canales en una instancia grande de Asterisk o 100 canales cada uno en cinco instancias diferentes. Esto hace posible construir soluciones del mundo real que se adaptan al tamaño de sus problemas.

Ejecución de grandes instancias de WombatDialer

Por supuesto, WombatDialer tiene que trabajar duro para manejar muchas llamadas a la vez.

Tiene que recuperar números y atributos de múltiples listas, borrarlos de las listas negras, hacer un seguimiento de las llamadas que se recuperan, y hacer un seguimiento de las llamadas en vivo y de las colas en vivo, ya que cada una de ellas envía sus propios eventos. Además, Wombat sincroniza el estado actual con una base de datos con bastante frecuencia, de modo que en caso de que algo vaya muy mal y se bloquee, puede reiniciar exactamente desde donde lo dejó sin perder ninguna llamada.

Esto es un gran trabajo, y ocurre en tiempo real.

Cuando se ejecuta el Wombat, hay algunos elementos que tienen que ser ajustados para asegurarse de que todo funciona sin problemas; a saber:

- Carga del sistema de I/O y de la base de datos
- Retardo de marcación y colas de comandos
- Retardo de la red
- Políticas de memoria y GC de la JVM
- Humanos!

Carga del sistema

Un sistema WombatDialer en funcionamiento sólo puede ser tan rápido como la base de datos con la que se sincroniza. Aunque WombatDialer utiliza la agrupación por lotes para enviar a la base de datos un conjunto eficiente de actualizaciones, en algún momento vas a notar que hay una espera significativa de I/O en el sistema.

No quieres que la espera de I/O sobrepase el 5-10%, porque esto hace que Wombat sea menos reactivo; tarda más en obtener datos y más en responder a los cambios en Asterisk, es decir, en hacer nuevas llamadas.

Esto se puede atenuar ajustando la cantidad de memoria disponible para que MySQL la utilice como caché; la configuración por defecto con la que viene MySQL es inadecuada para ejecutar miles de llamadas en paralelo.

Una capa de disco más rápida (usando SSDs y discos de datos separados) también ayuda mucho y puede marcar la diferencia.

Además, el Wombat tiene que obtener datos del disco para saber qué llamadas deben realizarse a continuación. Lo hace procesando lotes de llamadas, ya que esto es mucho más rápido que solicitar cada llamada por separado, esto significa que cuando se libera un canal, Wombat ya tiene una llamada en caché lista para ser colocada. Este valor es controlado por el parámetro Batch size en cada campaña. Un tamaño de lote más pequeño hará que Wombat haga más trabajo en la base de datos, pero hará que se añada menos latencia. Sugerimos utilizar un valor que sea aproximadamente el doble del número de canales posibles para una campaña determinada.

Latencia de marcación

La latencia de marcación es la cantidad que tarda Asterisk en "confirmar" una llamada después de haberla solicitado.

En un sistema que funcione correctamente, muy pocas llamadas deberían aparecer en la página Live en estado SOLICITADO. Esas son llamadas que han sido solicitadas a Asterisk pero que aún no han sido confirmadas.

La latencia de la marcación puede dividirse en dos partes:

- El tiempo que tarda una llamada solicitada en llegar a Asterisk.
Asterisk no puede procesar un conjunto infinito de solicitudes por segundo, y probablemente se colgará cuando se sobrecargue. Wombat intenta mitigar esto dividiendo la cola de comandos en "unidades de tiempo", donde sólo se pueden procesar tantos elementos en cada unidad. Por ejemplo, el valor por defecto es evitar enviar más de 5 comandos cada 50 milisegundos. Esta es una estimación conservadora, y dependiendo de tu hardware Asterisk un límite de 10 o incluso 20 mensajes cada 50 milisegundos podría funcionar para ti. Tener múltiples servidores de Asterisk significa que hay colas de mensajes separadas que se conectan a cada instancia de la PBX y por lo tanto cada cola sólo contiene una fracción de todas las llamadas salientes.
- El tiempo que tarda Asterisk en confirmar que se está procesando una llamada solicitada.
Si este tiempo empieza a estar en el orden de magnitud de los segundos, es probable que el sistema esté sobrecargado. Está bien que los tiempos de confirmación aumenten cuando se inicia una campaña con muchos canales que llenar a la vez; pero no está bien que sean altos de forma constante todo el tiempo.

Puedes hacer un seguimiento del tiempo combinado mirando el valor llamado Wait Pre en los informes, y observando el porcentaje de llamadas en estado SOLICITADO en la página Live. Vigile los tiempos máximos y medios de "Pre Espera" en sus campañas.

La capa de red puede añadir una latencia de red adicional; WombatDialer está diseñado para funcionar mejor con una base de datos local y servidores Asterisk locales.

El tiempo de respuesta a la base de datos y a los servidores Asterisk debería ser idealmente de cero milisegundos; los valores superiores a 100 milisegundos (por ejemplo, ejecutar WombatDialer en los Estados Unidos con servidores Asterisk en Europa y una base de datos en Singapur) no funcionarán para escenarios de alta carga.

Además, el ancho de banda entre todas las máquinas debe ser de clase data-center o LAN; considere que cada instancia grande de Asterisk puede generar 4-5 megabits de eventos por segundo.

Memoria y GC

En cuanto a la memoria, WombatDialer no necesita realmente mucha memoria para funcionar - puedes tener el motor procesando miles de llamadas paralelas en tan solo 256M de heap.

Dicho esto, si ejecutas informes, tienes varios usuarios accediendo al sistema, mantienes la página Live abierta y cargas nuevas listas de llamadas al mismo tiempo, la cantidad de RAM necesaria puede ser significativamente mayor. Además, como en todas las aplicaciones Java, tener menos memoria disponible significa que la JVM tiene que ser más agresiva en las recolecciones de basura.

Esto lleva a una demora muy desigual y puede incluso causar eventos de recolección de basura de "parar el mundo". Esto es algo que quieres evitar a toda costa; la memoria es muy barata, por lo que debes dar a Wombat una buena cantidad de RAM para que funcione, y utilizar un recolector de basura generacional orientado al rendimiento que esté optimizado para evitar eventos de recolección importantes al precio de usar más memoria de trabajo.

Patrones de uso

El último punto de nuestra lista son los patrones de uso humano; cuanto menos actividad tenga el sistema en las horas punta de marcación, mejor.

No dejes la página Live abierta "por si acaso". Trata de evitar ejecutar informes importantes o cargar listas muy grandes cuando tu sistema esté completamente cargado - lo más probable es que puedas hacerlo en otro momento. WombatDialer incluye una API muy amplia que permite un control remoto completo de las instancias de Wombat.

Finalizando

Dicho esto, es razonable ejecutar miles de llamadas paralelas en un sistema WombatDialer con un hardware moderno y de alta gama.

A lo largo de los años, hemos realizado algunos ajustes y optimizaciones importantes en Wombat, lo que significa que WombatDialer es aproximadamente el doble de eficiente que las versiones anteriores.

En cualquier caso, recuerda hacer una prueba de carga importante antes de pasar a producción. Tanto si tienes 100 como 10.000 canales, no querrás toparte con un muro el día de la puesta en producción.

Puede que marcar miles de canales no sea tu pan de cada día, pero es bueno saber que puedes hacerlo con sólo pulsar un botón. Creemos que una buena herramienta debe darte márgenes de crecimiento y preservar tu inversión y experiencia actuales. Trabajamos duro para que no tengas que preocuparte por ello.

Referencias de WombatDialer

Para obtener más información técnica sobre la solución de centro de llamadas WombatDialer, consulte el [Manual del Usuario](#).

Visita <https://www.wombatdialer.com/> para una prueba completa.

Asista a nuestros [Free Webinars](#) para ver una demostración en directo de QueueMetrics..