

Marcación eficiente inigualable

Tu **WombatDialer** es una herramienta bastante poderosa: permite ampliarse lo suficiente para poder gestionar **miles de llamados** al mismo tiempo. Sin importar si tienes 100 o 10.000 canales, no querrás encontrarte con una piedra en el camino el día que implantes la producción.



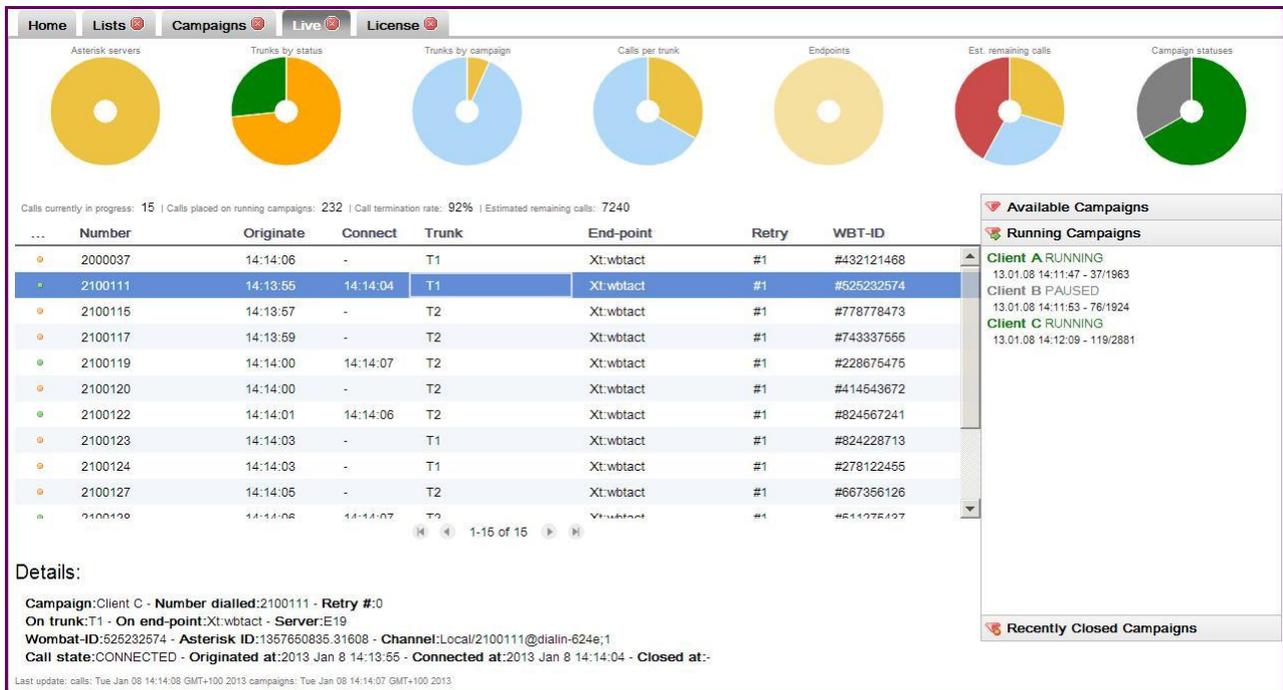
Primero, definamos qué queremos decir con **marcación de "alto volumen"**.

Con el fin de redactar este artículo, definimos "alto volumen" a una cantidad de llamadas que saturarían a un sistema Asterisk. Es difícil darle un valor específico, ya que depende del tipo de hardware que uses y si qué es lo que realmente hace tu PBX (centralita para extensiones) cuando procesa las llamadas. Si de verdad deseas darle un valor, entonces pueden llegar a ser unas 250 llamadas, más o menos 150.

Optimizar Asterisk

Ya que Wombat confía en Asterisk para la marcación, es importante comprender cuáles son los desafíos de procesar un alto volumen de llamadas en tu sistema Asterisk:

Primero, lo que realmente importa. La cantidad de llamadas que tu PBX puede gestionar con éxito depende de qué es lo que haces y cuán cara es la llamada. Existen tres supuestos comunes en este caso: decodificación, grabación de llamadas y secuencias AGI.



La decodificación usará mucho CPU (unidad central de procesamiento) para traducir el audio de un formato a otro. Esto ocurre en silencio cada vez que se reproduce una grabación en un formato y tu canal lo usa en otro formato. La música en espera es en MP3 y los canales transmiten G729, eso significa la decodificación. Aunque esta es una característica muy cómoda que ofrece Asterisk, no es lo correcto para las marcaciones de alto volumen.

Las grabaciones de llamadas pueden volverse muy caras pronto, ya que usarán IOPS precioso de disco para escribir cientos de archivos de audios en ciclos de disco y de CPU valiosos para convertir audio en tu formato de almacenamiento favorito. Si realmente necesitas grabar llamadas, entonces debes asegurarte de que esta función esté encendida sólo cuando es requerido (por ejemplo, una llamada entrante con un agente), que los discos sean tan rápidos como los puedes obtener (un ramdisk es una buena opción en este caso) y que el formato de almacenamiento requiera el menor valor de CPU posible. Para las grabaciones a gran escala, una buena opción puede ser utilizar un grabador pasivo separado (red).

Las secuencias AGI requieren que se ejecute en proceso en cada caso. A menudo se usan en Asterisk, ya que ofrecen una forma simple de "adherir" un porcentaje de lógica y acceso a servicios externos (servicios web o bases de datos) al plan de marcación Asterisk. Por lo general están escritos en lenguajes de secuencia lentos e interpretados y

definitivamente no escalarán. Al ejecutar Wombat, siempre suele ser una mejor opción recolectar con anterioridad cualquier valor de interés que provenga de sistemas externos y que Wombat los pase a Asterisk como una variable de canal. Por ejemplo, en lugar de ejecutar una secuencia AGI que ejecuta un intérprete que abre una conexión de base de datos sobre la cual ejecutas una búsqueda SQL para decidir si deberías 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 a reproducir.

Lo segundo que se debe considerar es **si lo indicado para ti es un sistema con base GUI**. Aunque contar con GUI hace que tu vida sea más fácil en cuando a la administración del sistema, el tipo de plan de marcación que se genera a menudo es extremadamente complicado, por lo tanto lento. Aunque esto se hace para ofrecerte muchas características interesantes para tu oficina PBX, puedes darte cuenta que realmente no las requieres a todas cuando realizas marcaciones salientes a gran escala. Lo mejor de dos mundos es usar algún plan de marcación con codificación manual para hacer llamadas salientes (la parte que realiza Wombat) y usar tu GUI favorito para todas las otras funciones.

Una tercer área donde se debe hacer hincapié es la **interfaz AMI** que usa Wombat para operar Asterisk. El protocolo AMI puede volverse muy locuaz, ya que el sistema genera muchos eventos, y esto puede llevar a una latencia importante antes de que se procesen los eventos. Esto se puede abarcar al filtrar el juego de eventos que Asterisk envía (por ejemplo: Wombat no necesita eventos de ejecución de plan de marcación), pero inclusive en este caso en algún punto el problema se manifestará por sí solo. Además, Asterisk sólo puede procesar tantas solicitudes por segundo, y en algún punto querrás generar muchas llamadas para que gestione el sistema Asterisk.

Un último problema general con Asterisk es que **éste procesa canales de voz**; si el sistema está sobrecargado y la voz no se procesa a tiempo real, tendrás **problemas de calidad de voz**. Realizar llamadas y ofrecerles un mensaje de audio partido o incomprensible es algo que realmente deseas evitar. Es por esto que debes asegurarte de contar con la suficiente "capacidad de búfers" en tus sistemas para que la calidad de voz permanezca clara en todos los casos.

La verdadera solución para las marcaciones de alto volumen es realizar una **escala horizontal**. Esto significa operar múltiples servidores Asterisk en paralelo, cada uno de los cuales procesa sólo una cantidad limitada de canales que permanece bien dentro de sus límites. Por suerte, Wombat fue diseñado desde el principio para **manejar una gran variedad de servidores Asterisk como si fueran una máquina más grande y única**. Para Wombat casi no existe diferencia al procesar 500 llamadas en un caso superior o

100 canales en cinco instancias diferentes. Esto hace que sea posible construir soluciones reales para escalar y cubrir tus problemas.

Operar instancias grandes de WombatDialer

Por supuesto que Wombat tiene que trabajar duro para poder gestionar tantas llamadas al mismo tiempo. Debe recuperar números y atributos para listas múltiples, limpiarlas en contra de las listas negras, llevar un registro de las remarcaciones, y rastrear las llamadas en vivo y las filas en vivo, a medida que cada una envía sus propios eventos. Además, Wombat sincroniza el estado actual con una base de datos bastante a menudo, así que en el caso que algo saliera muy mal y dejara de funcionar, se puede reiniciar desde ese punto sin tener que perder ninguna otra llamadas. Esto es un gran trabajo, y ocurre a tiempo real.

Al ejecutar Wombat, existen algunos ítems que se deben afinar para asegurar de que todo opere bien, específicamente:

- Sistema entrante y saliente y carga de base de datos
- Latencia de marcación y filas de comandos
- Latencia de red
- Memoria JVM y políticas GC
- ¡Humanos!

Carga del sistema

Cuando se ejecuta un sistema WombatDialer, éste sólo puede ser tan veloz como la base de datos con la que se lo sincroniza. Aunque Wombat usa tandas para enviarle a la base de datos un juego de actualizaciones inteligente, en algún punto observarás que hay una espera importante de llamadas entrantes y salientes en el sistema. **No deseas que estas llamadas esperen más de un 10%**, ya que esto hace que Wombat sea menos reactivo; se requiere de mayor capacidad para recuperar la información y más para responder a los cambios en Asterisk, específicamente poder realizar nuevas llamadas. Esto puede atenuarse al afinar la cantidad de memoria disponible para que MySQL use como caché; los ajustes por defecto que MySQL contiene son inadecuados para operar miles de llamadas en simultáneo. **Una capa de disco más veloz** (que use SSD y discos de información separados) también ayuda mucho y puede marcar una diferencia.

Además, Wombat debe recolectar información del disco para poder saber cuáles son las próximas llamadas. Esto se realiza al procesar las tandas de llamadas, ya que esta forma es más rápida que solicitar cada llamada por separado. Esto implica que cuando una llamada se libera, Wombat ya tiene una llamada siguiente lista. Este valor se ve controlado por el parámetro "*Batch size*" en cada campaña. Un tamaño de carga menor hará que Wombat haga más trabajo de base de datos, pero hará que se pueda agregar

menos latencia. Sugerimos usar un valor que sea 2x de veces la cantidad de canales posibles para una campaña dada.

Latencia de marcación

La latencia de marcación es lo que requiere Asterisk para "confirmar" una llamada luego de que haya sido solicitada. En un sistema que opera bien, deberían aparecer muy pocas llamadas en la página En Vivo como SOLICITADAS. Esas son llamadas que han sido solicitadas a Asterisk pero todavía no han sido aceptadas. La latencia de marcación puede dividirse en dos partes:

-El tiempo que le lleva a una llamada solicitada llegar a Asterisk. Éste no puede procesar una cantidad infinita de solicitudes por segundo, y probablemente termine por dejar de funcionar cuando se vea sobrecargado. Wombat intenta mitigar esto al dividir la fila de comandos en "unidades de tiempo", donde sólo cierta cantidad de ítems pueden ser procesados en cada unidad. Por ejemplo, el valor por defecto es evitar enviar más de 5 comandos cada 50 milisegundos. Este es un estimativo conservador, y según el hardware de Asterisk tendrás un límite de 10 o 20 mensajes cada 50 milisegundos. Contar con múltiples servidores de Asterisk significa que existen filas de mensajes separadas que se conectan a cada instancia PBX y de esa forma cada fila sólo se ocupa de una fracción de llamadas salientes.

-El tiempo que le lleva a Asterisk para confirmar que una llamada solicitada está siendo procesada. Si este tiempo comienza a estar en el orden de magnitud de segundos, es posible que el sistema se sobrecargue. Está bien que los tiempos de confirmación alcancen su máximo cuando comienza una campaña en muchos canales al mismo tiempo; lo que no está bien es que el caudal sea alto todo el tiempo.

Puedes llevar un registro del tiempo combinado al mirar el valor que se llama "*Wait Pre*" en los informes, y al observar el porcentaje de llamadas en estado SOLICITADAS en la página En Vivo. Préstale atención a los tiempos máximos y promedio en tus campañas.

Es posible que la red agregue una **latencia de red** adicional, Wombat está diseñado para trabajar mejor con una base de datos local y servidores de Asterisk locales. El cambio de tiempos tanto para la base de datos como para los servidores de Asterisk debería ser idealmente de cero milisegundos; los valores más altos que 100 milisegundos (por ejemplo, ejecutar WombatDialer en EE. UU. con servidores Asterisk en Europa y una base de datos en Singapur) no funcionará en escenarios de cargas grandes. Además, el ancho de banda entre las máquinas debería ser el centro de información o clase LAN; considera que cada instancia Asterisk grande puede generar 4-5 megabits de eventos por segundo.

Memoria y GC

En el caso de la memoria, WombatDialer no necesita realmente mucha para operar ejecutarse, puedes contar con el motor procesando miles de llamadas en paralelo en tan poco como 256 M de amontonamiento. Dicho esto, si ejecutas informes, tienes múltiples usuarios que acceden al sistema, mantienes la página En Vivo abierta y cargas nuevas listas de llamadas al mismo tiempo, la cantidad de RAM que necesitarás será muchísimo mayor. Además, al igual que con las aplicaciones Java, contar con menos memoria disponible significa que el JVM tiene que ser más agresivo en las recolecciones de datos sucios. Esto conlleva una latencia muy dispareja e inclusive puede causar casos de recolección de datos "donde todo se detiene". Esto es algo que querrás evitar a toda costa, la memoria es muy económica, así que debería proveerle a Wombat con una buena cantidad, y usar un *recolector generacional* optimizado para evitar casos de recolecciones mayores al precio de utilizar más memoria.

Patrones de uso

El último ítem en nuestra lista son los patrones de uso humano; cuanto menos sea la actividad con la que cuentes en un sistema al momento pico de marcaciones, mejor. No dejes la página En Vivo abierta "por si acaso". Intenta evitar ejecutar muchos informes o cargar listas muy grandes cuando el sistema está cargado por completo; seguramente puedas hacer eso sin problemas en otro momento. Wombat incluye un API muy amplio que permite instancias completas de control remoto de Wombat.

Cierre

Dicho todo esto, es razonable ejecutar miles de llamadas en paralelo en un sistema **WombatDialer 0.9 con hardware moderno y de calidad superior**. Cuando lanzamos Wombat 0.9 logramos algo muy preciso y óptimo, que significa que WombatDialer es *dos veces más eficiente* que las versiones anteriores.

En cualquier caso, recuerda hacer una **prueba de carga importante antes de iniciar la producción**. Ya sea que tengas 100 o 10.000 canales, no querrás echar todo a perder.

Marcar miles de canales puede que no sea fácil, pero está bueno saber que puedes hacerlo al presionar sólo un botón. Creemos que una buena herramienta debe ofrecerte márgenes para que puedas crecer y preservar tu inversión y experiencia existente. Trabajamos duro para asegurarnos de que tu no debas preocuparte por nada más.

¡FUERZA AL WOMBAT!