

**UNIVERSIDAD GABRIELA MISTRAL
FACULTAD DE INGENIERIA**

**CONTROL INTELIGENTE
DE OPERACIÓN EN
SERVICIOS INFORMÁTICOS**

Memoria para optar al título de
Ingeniero de Ejecución en Informática

Autor : Jorge Alvear Yévenes
Profesor Guía : Roberto Carú Cisternas
Profesor Integrante: Jorge Tapia Castillo

Santiago – Chile
Diciembre, 2015

Agradecimientos

El presente trabajo es el fruto del conocimiento adquirido a través de los años y en muchas formas está guiado y formado por las personas que influyeron positivamente en mi vida.

Cada una de estas personas forma parte de lo que soy y siento una tremenda gratitud hacia ellos.

Mis más profundos agradecimientos a mis padres, que forjaron mis primeros pasos como estudiante y como persona, entregándome los valores fundamentales; a mi familia, que es la principal motivación para todos mis logros y objetivos alcanzados.

El más grande de los agradecimientos a mi mujer Katherine, quien me ha apoyado permanentemente en mi vida y estudios

...y por supuesto, la más potente de las motivaciones que una persona pueda tener, que es la alegría de mis hijos Jorgito e Ignacio.

Agradezco a la universidad Gabriela Mistral, que me ha entregado las herramientas necesarias para poder obtener conocimientos superiores y a Roberto Carú, quien ha estado presente siempre para guiar y motivar el término de esta carrera.

INDICE

Agradecimientos.....	1
I. INTRODUCCION.....	5
1.1 <i>El problema actual</i>	7
1.2 <i>Objetivos de la investigación</i>	12
1.3 <i>Justificación</i>	13
1.4 <i>Hipótesis</i>	15
II. MARCO TEÓRICO.....	16
2.1 <i>Servicios informáticos</i>	17
2.2 <i>Componente</i>	18
2.3 <i>Carga Operativa</i>	20
2.4 <i>Métricas de servicio</i>	21
2.5 <i>Sistemas de control en la actualidad</i>	23
2.6 <i>Conocimiento asistido</i>	25
2.6.1 <i>Conocimiento artificial</i>	27
2.6.2 <i>Conocimiento experto</i>	28
2.6.3 <i>Madurez del conocimiento a través de la ciencia cognitiva</i>	29
III. DESARROLLO DEL TRABAJO.....	31
3.1 <i>Metodología</i>	31
3.2 <i>Planteamiento del problema</i>	31
3.3 <i>Posibles soluciones</i>	32
3.4 <i>Análisis matemático</i>	37
3.5 <i>Conocimiento asistido</i>	39
3.6 <i>Módulo de medición</i>	42
3.6.1 <i>Módulo de medición local</i>	42
3.6.2 <i>Módulo de medición remota</i>	46
3.7 <i>Módulo de Control Inteligente</i>	49
3.8 <i>Representación gráfica de la información</i>	58
3.9 <i>Gráfica de asistencia a la operación</i>	62
IV. HALLAZGOS	64
4.1 <i>Contrastar más de un período de tiempo</i>	64
4.2 <i>Generar alarmas de quiebre de umbrales</i>	64
4.3 <i>Agregar más zonas de comparación horaria</i>	64

4.4	<i>Registrar los procesos top 10 en quiebres de umbrales</i>	65
4.5	<i>Normalizar períodos anuales</i>	65
V.	CONCLUSIONES	66
5.1	<i>El cálculo matemático es posible</i>	66
5.2	<i>Es posible graficar la información de forma simple</i>	66
5.3	<i>Es posible graficar la información en tiempo real</i>	66
5.4	<i>Se comprueba la utilidad de la herramienta</i>	66
5.5	<i>Excelente herramienta de apoyo</i>	67
5.6	<i>Interpretación asistida en tiempo real</i>	67
	GLOSARIO	68
	BIBLIOGRAFÍA	71

INDICE DE FIGURAS

FIGURA 1 – EJEMPLO DE MONITOREO ACTUAL EN BASE A % DE OCUPACIÓN.....	8
FIGURA 2 – EJEMPLO DE ALZA REPENTINA EN USO DE RECURSOS.....	9
FIGURA 3 – DESCENSO REPENTINO EN USO DE RECURSOS	10
FIGURA 4 – ALZA DE CONSUMO, SIN ALCANZAR LÍMITE DE MONITOREO	12
FIGURA 5 – 3 NIVELES DE ANÁLISIS SOBRE SERVICIOS INFORMÁTICOS.....	17
FIGURA 6 – COMPOSICIÓN DE SERVICIOS.....	18
FIGURA 7 – COMPONENTES EN UN SERVIDOR	19
FIGURA 8 – USO PORCENTUAL DE RECURSOS.....	20
FIGURA 9 – USO PORCENTUAL DE RECURSOS EN TRANSURSO DE TIEMPO	21
FIGURA 10 – USO PORCENTUAL DE CPU POR TIEMPO	22
FIGURA 11 – USO PORCENTUAL DE RECURSOS Y SU UMBRAL DETERMINADO	24
FIGURA 12 – ÍCONO DE CONOCIMIENTO ASISTIDO	26
FIGURA 13 – ZONAS HORARIAS DE ESTUDIO	33
FIGURA 14 – MONITOREO DE USO DE CPU CON LÍMITE INFERIOR Y SUPERIOR.....	35
FIGURA 15 – MONITOREO DE USO DE MEMORIA CON LÍMITES INFERIOR Y SUPERIOR.....	35
FIGURA 16 – RANGOS CALCULADOS (COMPARACIÓN HISTÓRICA Y ACTUAL).....	36
FIGURA 17 – CÁLCULO DE LA VARIANZA.....	37
FIGURA 18 – CÁLCULO DE LA DESVIACIÓN ESTÁNDAR	38
FIGURA 19 – EXPLICACIÓN DE RANGO MEDIDO Y SUS UMBRALES (MÍNIMO, MÁXIMO).....	38
FIGURA 20 – EXTENSIÓN DE UMBRALES DE ACUERDO A INFORMACIÓN HISTÓRICA	39
FIGURA 21 – NIVELES DE OPERACIÓN (CPU) QUIEBRAN EL UMBRAL SUPERIOR	41
FIGURA 22 – SISTEMA DE MONITOREO CENTREON.....	59
FIGURA 23 – CENTREON CONTROLANDO UN SERVIDOR DE PRODUCCIÓN.....	60
FIGURA 24 - GRÁFICA INDIVIDUAL DEL TÓPICO MEMORIA, RECOLECTADA DE FORMA CENTRAL Y EXPRESADA CON SUS ESTADOS.....	60
FIGURA 25 - GRÁFICA INDIVIDUAL DEL TÓPICO MEMORIA, RECOLECTADA DE FORMA CENTRAL Y EXPRESADA CON SUS ESTADOS.....	61
FIGURA 26 – DESVIACIÓN ESTÁNDAR X SEGMENTO HORARIO.....	62
FIGURA 27 - MÉTRICAS PARA CONTROL INTELIGENTE DE CPU.....	63
FIGURA 28 – MÉTRICAS PARA CONTROL INTELIGENTE DE MEMORIA.....	63

I. INTRODUCCION

En la actualidad, los sistemas informáticos prestan servicios que están sometidos a estrictas reglas de negocio, con una obligación de disponibilidad que bordea el 100%.

Casos de ejemplo en talla empresarial, como bancos, líneas aéreas, retail, etc, tienen la obligación de ofrecer una disponibilidad de los sistemas cercana al 99.89, es decir, se permite poco menos de 9 horas de indisponibilidad al año.

Variadas son las causas que generan indisponibilidad, entre las cuales podemos seleccionar: mala intervención (falla humana), fallas de infraestructura (equipos físicos), servicios no controlados (servidores no monitoreados), errores en las aplicaciones y fallas por falta de capacidad.

Evitar incidentes y así mantener una disponibilidad aceptable es tarea fundamental de las áreas de informática y producción, lo que implica tener un control exhaustivo de los servicios informáticos y un conocimiento profundo del comportamiento de cada uno de ellos.

Las tareas centrales en la operación diaria, son minimizar la cantidad de eventos de indisponibilidad (pérdida de servicio), detectar los riesgos de esta, mitigarlos y coordinar procesos de mejora continua, con el objetivo de desarrollar un ciclo virtuoso con respecto al conocimiento obtenido de la plataforma y sus servicios.

Los equipos humanos que administran estas plataformas tecnológicas, deben desarrollar una serie de habilidades y conocimientos para reducir los incidentes con pérdida de servicio, pero el proceso de madurez es lento y no es asistido por herramientas tecnológicas que ayuden a un control efectivo.

El presente trabajo está orientado a lograr determinar si es posible asistir o acelerar el conocimiento que se puede obtener del comportamiento de servicios informáticos y cuáles serían los mejores medios técnicos para lograrlos.

1.1 El problema actual

El apoyo más común para estos equipos, es un monitoreo de los niveles de operación en la plataforma, pero está más enfocado en los límites de capacidad, que en determinar cuál es el nivel de operación normal proyectado para un instante de tiempo, basado en el conocimiento experto y datos históricos.

De esta forma y como no existe el conocimiento real de los niveles de operación esperados para una jornada, lo habitual es establecer umbrales de forma arbitraria, que generan alertas en caso de que el servicio exceda estos parámetros.

Usualmente, estos parámetros de alerta son:

Advertencia 80% de ocupación de capacidad

Estado Crítico 90% de ocupación de capacidad

Un ejemplo del monitoreo actual en la siguiente imagen, muestra el porcentaje de ocupación del recurso CPU en un servidor determinado, marcando el límite de capacidad de operación normal en el 80%.

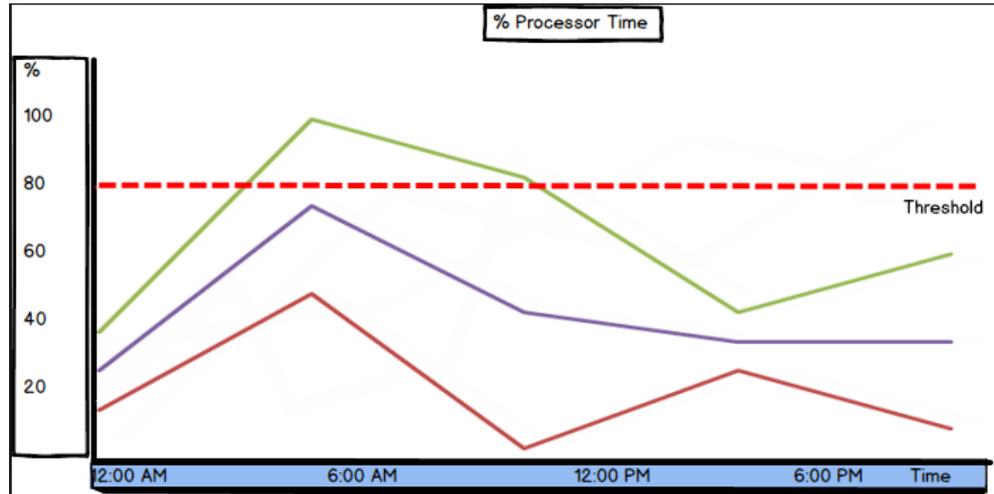


Figura 1 – Ejemplo de Monitoreo Actual en base a % de ocupación

Los equipos humanos proceden a una revisión de los componentes, sólo si es gatillada una de estas alertas y generalmente necesitarán el apoyo de grupos expertos o con conocimiento de las aplicaciones y servicios, para determinar si se trata de una situación normal o corresponde a un problema (por ejemplo un ataque informático o una degradación en la respuesta de Base de Datos).

Esto significa, que si los umbrales no se han excedido a pesar de haber tenido un aumento sustancial de carga, el servidor quedará fuera de una revisión.

En casos donde el parque de servidores supera los miles, no será posible entender los niveles de operación particulares, dejando fuera del análisis todas las variaciones importantes.

Lo anterior deja fuera del análisis aspectos realmente importantes a la hora de analizar las variaciones de carga y a continuación expondré ejemplos que son frecuentes dentro de los servicios TI.

Ejemplo 1

Aumento sustancial del consumo de recursos en un sistema que ha tenido regularidad.

Un servidor web ha tenido un nivel de carga regular en los últimos 12 meses, con índices de operación que fluctúan entre el 30% y el 33% de consumo de CPU. De pronto, el servidor pasa a tener un consumo de CPU del 75%. Como el consumo de CPU no ha quebrado ningún umbral de alerta (80% warning – 90% critical), no será analizado por un experto y peor aún, no se tomará registro de ningún aspecto de servicio para su posterior análisis.

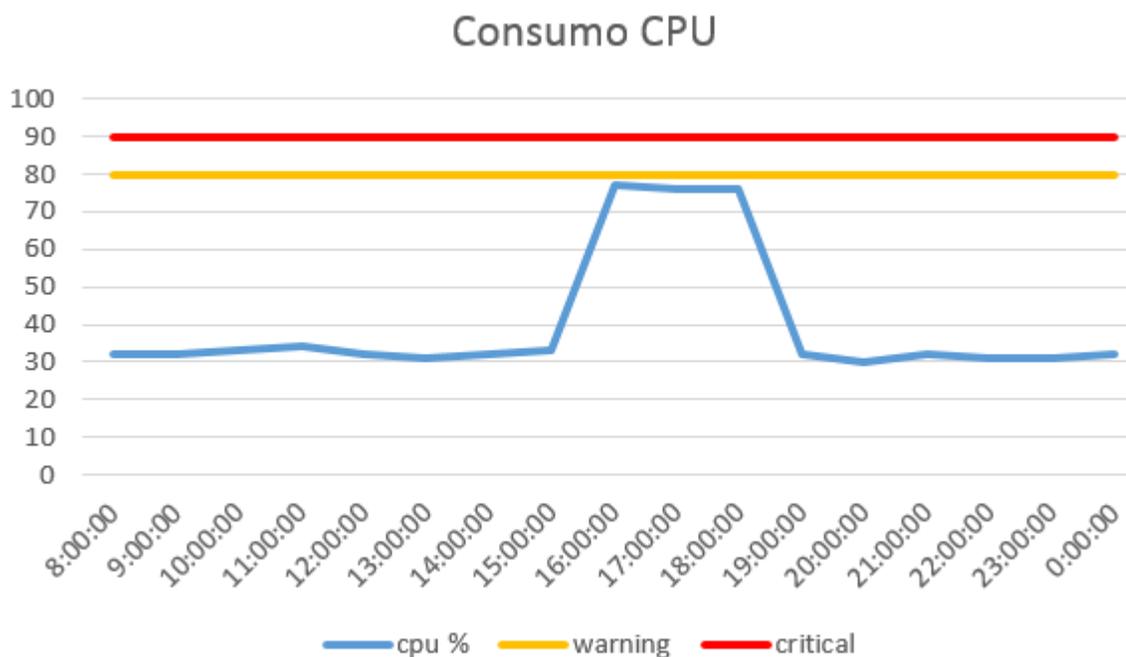


Figura 2 – Ejemplo de alza repentina en uso de recursos

Ejemplo 2

Baja sustancial de los niveles de consumo.

Un servidor de base de datos que es importante para la empresa y que tiene la función de almacenar todos los datos de ventas del mes para su análisis posterior por parte del área comercial, tiene un consumo promedio del uso de memoria del 70% durante el día y del 40% durante la noche. En un instante, el servidor pasa a tener un uso de memoria del 10% de forma constante en el día y noche. Esta situación también quedará fuera del análisis experto, porque no ha quebrado ningún umbral de alerta, siendo que es bastante probable que no esté registrando todos los datos esperados, lo que generará problemas posteriores a la hora de ser requeridos para el análisis comercial.

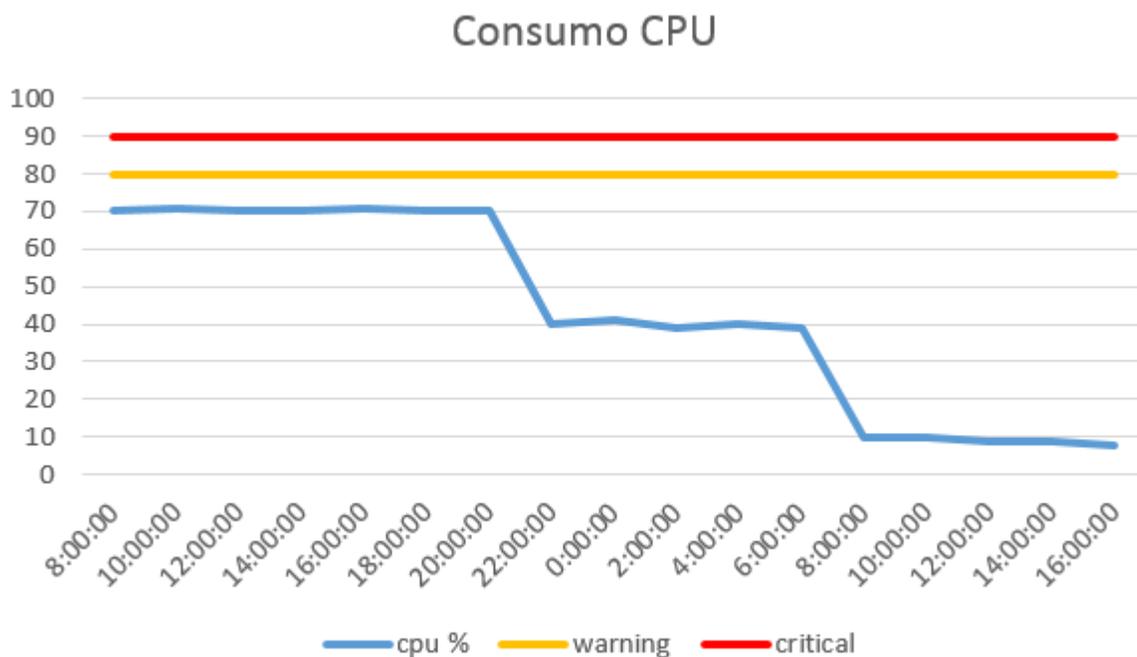


Figura 3 – Descenso repentino en uso de recursos

Ejemplo 3

Un servicio TI en Chile, ha tomado la administración de 10 servidores de una empresa hotelera en Brasil.

Estos servicios se mantienen con un nivel de carga de procesamiento estable, con fluctuaciones entre un 50% y un 60% todo el día. Durante la primera semana, se ve un aumento en el consumo de los servidores de venta, que está quebrando los umbrales de alerta día a día. Los técnicos informáticos especialistas no han encontrado situaciones irregulares a nivel tecnológico, por lo que requieren del análisis de expertos de la empresa hotelera.

Después de un análisis detallado, han concluido que la causa es una alta demanda previa al feriado de Tiradentes (propia de ese país), para el cual existen promociones vigentes. Es posible que no exista un registro técnico de este comportamiento, por lo cual al año siguiente, otros administradores informáticos se verán enfrentados a la misma situación, sin que el conocimiento detallado de este incidente les sea alertado de forma previa y tendrán que repetir el mismo ciclo de análisis o bucear en la información registrada por analistas anteriores, pero que sólo se encuentra almacenada y no está expuesta por alguna herramienta técnica.

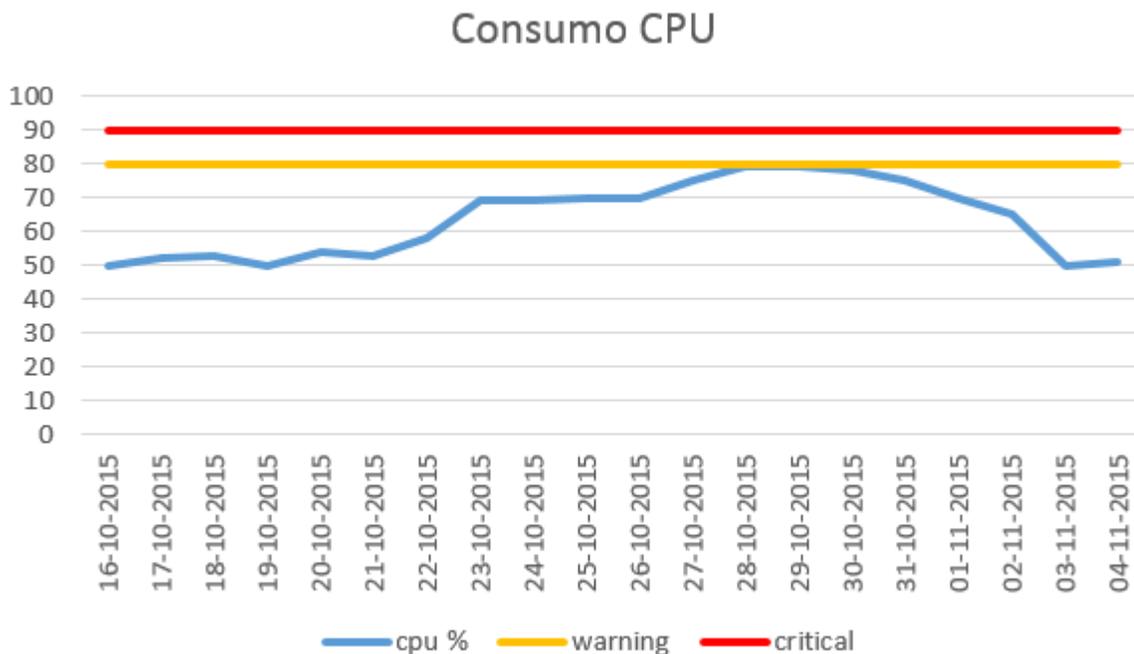


Figura 4 – Alza de consumo, sin alcanzar límite de monitoreo

1.2 Objetivos de la investigación

Como se ha expuesto anteriormente, el objetivo del presente trabajo es determinar la posibilidad de lograr un conocimiento a través del análisis y estudio sistemático de los niveles de operación históricos de una plataforma informática.

El estudio se hará sobre los datos recolectados para los componentes tecnológicos más comunes en el área de informática, que son los servidores.

El consumo de recursos, se deberá enfocar en los 2 tópicos más comunes de estos componentes, que son el uso de CPU y la utilización de Memoria.

La investigación estará centrada en 3 objetivos principales:

- 1- Comprobar la posibilidad de pronosticar los niveles de consumo esperados para una jornada de operación, a través del conocimiento asistido.
- 2- Encontrar los medios técnicos y los datos necesarios para determinar patrones de comportamiento y asistir el conocimiento de los grupos humanos.
- 3- Buscar una forma apropiada para presentar la información obtenida y exponer en línea los niveles esperados para una reacción o prevención oportuna.

1.3 Justificación

El trabajo y la investigación están plenamente justificados desde diferentes puntos de vista, tanto con la motivación de poder apoyar el control de las plataformas tecnológicas y de poder ir creando un conocimiento de forma asistida respecto del uso de los componentes.

- La importancia que ha cobrado actualmente mantener la continuidad operacional de los sistemas informáticos, debido a que encontramos en casi cualquier ámbito, la necesidad de operar ininterrumpidamente.

- La importancia que ha tomado el rol de los sistemas informáticos en cualquier organización y por lo tanto la relevancia de los datos y sistemas que en él existen.
- La necesidad de acelerar el conocimiento que los grupos humanos van generando de los servicios informáticos, principalmente aquellos servicios llamados de outsourcing TI, en el cual se externaliza su administración y control, con compromisos de operación que se deben cumplir desde el primer día.
- Disminuir los incidentes de indisponibilidad causados por situaciones de carga no esperados o prevenir situaciones de fluctuación mayor, con información oportuna, que permita anticipar acciones, lo que generalmente tiene una repercusión positiva en la reducción de costos.
- Reducir el impacto que la indisponibilidad de servicios tiene en los negocios que operan online, tanto en aspectos financieros, como en la imagen hacia los usuarios finales, que requieren un servicio estable y siempre disponible.

También es imprescindible mencionar que debido a los controles implementados en la actualidad, se genera un volumen de datos importantes respecto al comportamiento de cada servicio y componente de informática, que no son explotados y que podrían ayudar fuertemente a la labor humana sobre los sistemas.

1.4 Hipótesis

Se plantea en el presente trabajo, la hipótesis de que es posible obtener un conocimiento, respecto al comportamiento de uso de un servicio informático, utilizando las métricas históricas y lograr representarlas de una forma sencilla para facilitar su interpretación.

Esto le puede dar una ventaja a los servicios en la predicción de problemas, utilizando información valiosa, que actualmente sólo es almacenada.

El estudio debe enfocarse principalmente en 3 aristas:

- Determinar la posibilidad de un cálculo matemático que interprete de forma correcta los límites de operación, usando la estadística de dispersión en los datos obtenidos en el pasado.
- Construir el medio técnico para extraer y procesar la información histórica, entregando el cálculo de límites de operación y sus umbrales en tiempo real.
- Obtener un medio técnico, para mostrar la información de forma simple y en tiempo real.

El enfoque del trabajo será poder estudiar soluciones y elegir la mejor para cada caso. Como el sistema aún no está diseñado, se debe contemplar la posibilidad de generar simulaciones con datos estadísticos de muestras obtenidas de ambientes informáticos reales y productivos.

II. MARCO TEÓRICO

El planteamiento del presente trabajo, es determinar la posibilidad concreta de asistir el conocimiento experto de los equipos humanos, respecto a los niveles de operación normal en un determinado servicio informático, con la finalidad de poder tener un pronóstico certero de sus límites en un instante de tiempo presente.

Esto permitiría un control de la carga operativa, basado en antecedentes históricos de comportamiento en cada uno de los componentes que integran un servicio informático.

En la actualidad, para casi todo componente informático que interviene en procesos productivos, existe un control de operación y una medición de su comportamiento, lo que genera registros detallados a través del tiempo.

Estos registros son muy poco explotados y actualmente son usados sólo como un medio de análisis para determinar si la capacidad total del servicio está llegando a su límite o puede permanecer en funcionamiento por otro período de tiempo.

Los problemas de operación, sólo son estudiados una vez que los niveles de consumo de estos recursos traspasan umbrales pre-establecidos, de acuerdo a los límites de capacidad que el componente tiene.

2.1 Servicios informáticos

Se llama servicio informático a un grupo de componentes de tecnología destinados a una función común (o tarea común) dentro de un proceso o modelo de negocio. (Larry Klosterboer - IBM, 2011)

En la escala de orden de las plataformas tecnológicas, es la agrupación que se encuentra entre un componente (component) y un negocio (business).

A través de las mediciones de componentes comunes para un servicio informático, se puede determinar los niveles de operación y el estado de esta, en un término que se denomina estable o sano (healthy).

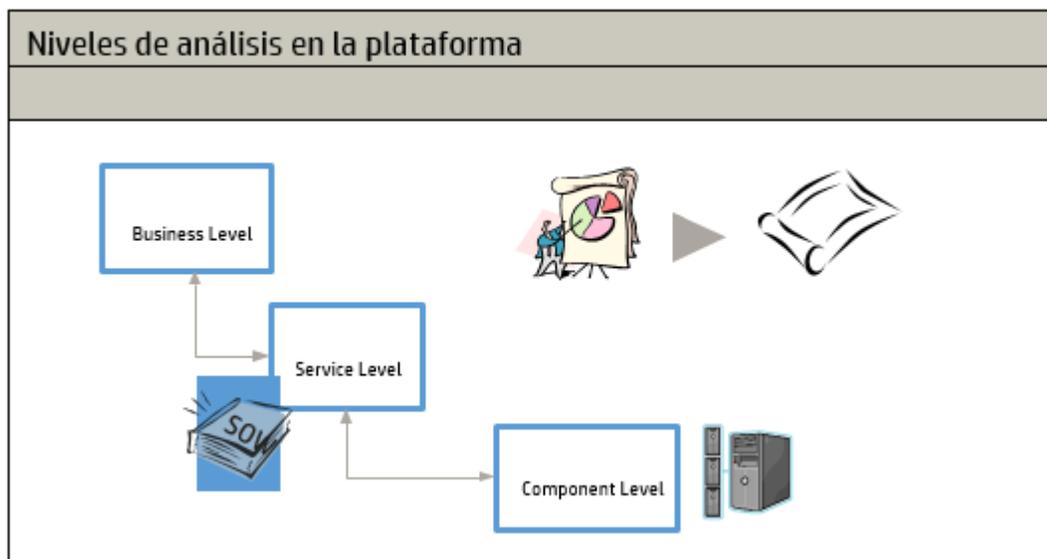


Figura 5 – 3 niveles de análisis sobre servicios informáticos

2.2 Componente

Se llama componente a cualquier artefacto tecnológico que desempeñe una tarea dentro de un servicio informático. (Larry Klosterboer - IBM, 2011)

Por ejemplo, si tomamos como referencia un sistema de ventas online, uno de sus servicios informáticos sería los servidores web. Estos funcionan alojados en servidores de aplicación, routers de comunicaciones y balanceadores de carga.

Cada uno de estos se considera un componente de infraestructura que tienes recursos o tópicos interesantes de ser medidos.

En el caso de los componentes que necesitamos estudiar en su comportamiento, escogeremos los que están más relacionados con el procesamiento, que son los servidores.

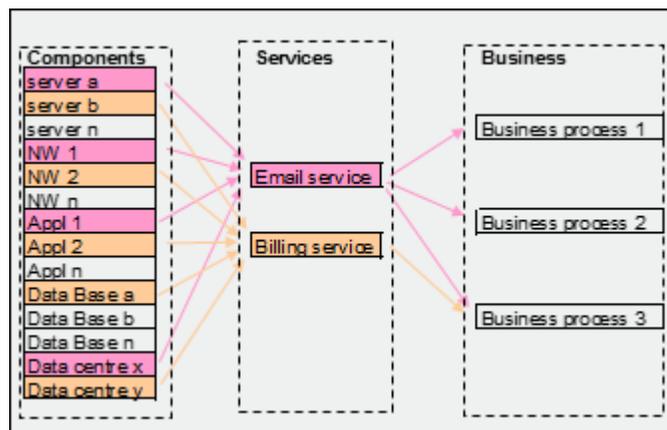


Figura 6 – Composición de servicios

Los servidores tienen varios tópicos interesantes de medir:

- Uso de CPU
- Uso de Memoria
- Uso de Disco
- Uso de Red



Figura 7 – Componentes en un servidor

Para el caso del presente trabajo, nos enfocaremos en los 2 de mayor importancia a la hora de determinar comportamiento y niveles de carga, que son el uso de CPU y Memoria.

La utilización de estos recursos, se mide y registra en términos de porcentaje de uso del recurso, que será la forma en que procesaremos la información.

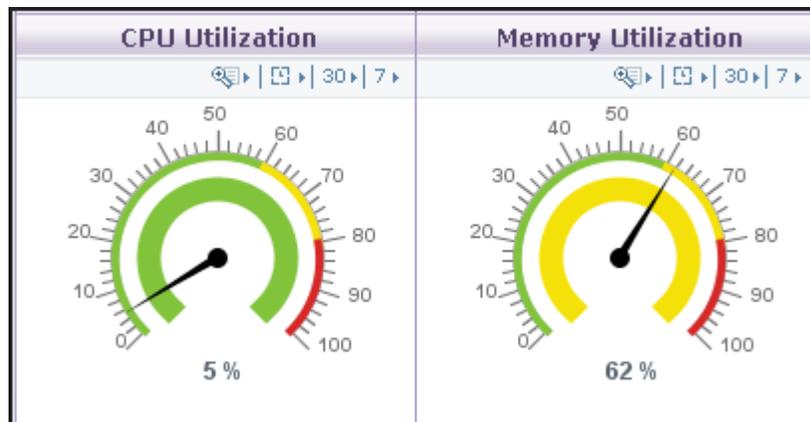


Figura 8 – Uso porcentual de recursos

2.3 Carga Operativa

Se determina como carga operativa al porcentaje de uso de la capacidad de un componente en un instante de tiempo. (Allison Kuzenko - HP, 2015)

En la informática, la medición y control de la carga operativa son la tarea central de los equipos de producción, pues es el esfuerzo más importante para mantener un servicio estable, reducir los riesgos de indisponibilidad y aprovisionar recursos oportunamente.

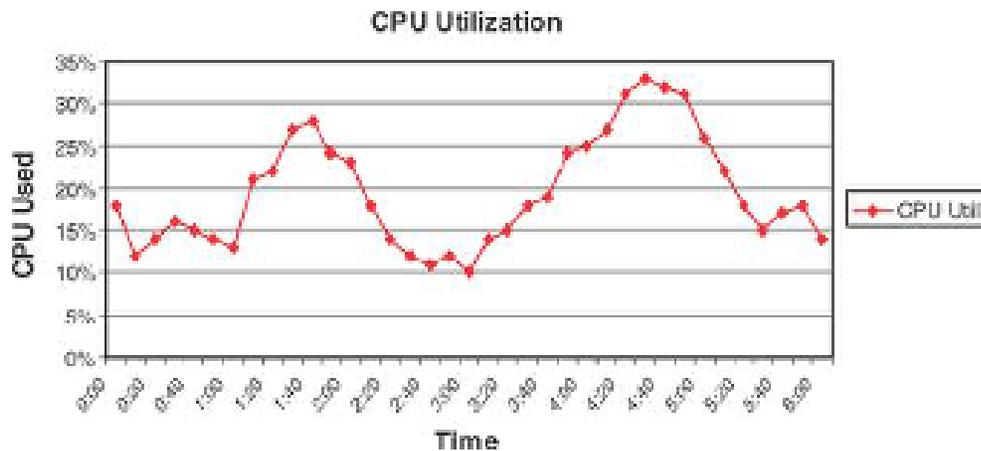


Figura 9 – Uso porcentual de recursos en transcurso de tiempo

El presente trabajo usará como ejemplo para el método experimental las métricas de consumo de servidores, con los datos registrados históricamente de procesamiento (CPU) y memoria (MEM).

Un hallazgo que puede ser abordado como complemento del presente trabajo, pueden ser el consumo de disco y otros parámetros importantes como cantidad de usuarios o cantidad de procesos para un dueño específico.

2.4 Métricas de servicio

Las métricas de servicio son valores numéricos que representan el porcentaje de uso de un componente en un instante de tiempo.

Trabajaremos con métricas que se denominan como utilización porcentual de recursos, específicamente con los de CPU y Memoria.

Como ya se ha dicho antes, actualmente sólo se preparan controles acordes a determinar que la operación no sobrepase límites de la capacidad del componente medido, usualmente con los umbrales de 80% para una advertencia (warning) y 90% para determinar un posible problema de sobreconsumo (estado crítico).

A continuación un ejemplo de lo que encontramos usualmente en las mediciones de componentes a través de su monitoreo estándar. Porcentaje de uso en función del tiempo.

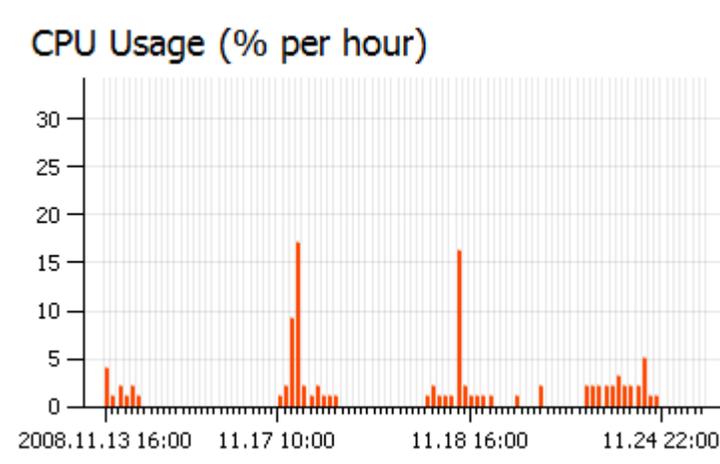


Figura 10 – Uso porcentual de CPU por tiempo

2.5 *Sistemas de control en la actualidad*

Actualmente, los sistemas de control de carga que operan sobre los equipos tecnológicos, están destinados a medir el comportamiento en términos de ocupación de la capacidad, lo que permite mantener una operación dentro de márgenes previamente establecidos y dentro de límites considerados sanos (healthy).

Estos límites son fijados de forma subjetiva y como se describe en la introducción, están enfocados en la capacidad de los sistemas más que en determinar el estado normal en un instante de tiempo.

Así por ejemplo, lo normal es que si no existe conocimiento experto de los niveles de carga en un servicio, se procede a fijar límites del 80% para un nivel de advertencia y 90% para un nivel crítico.

El problema de este control es que quedarán fuera del análisis aspectos aún más importantes, como por ejemplo un aumento del 50 en el uso de procesador después de 2 años de mantenerse en un 20%.

En este caso teníamos un sistema estable, con una ocupación promedio del 20% de procesador y de un momento a otro varía a un 70%, pero quedará fuera de la revisión de un experto porque no ha cruzado ningún umbral de alerta.

De la misma forma, otro ejemplo clásico es la baja sustancial en el uso de algún componente, así por ejemplo, si un servidor baja su uso de memoria que ha permanecido estable en un 50%, baje de pronto a un 5%, quedará fuera de una revisión experta.

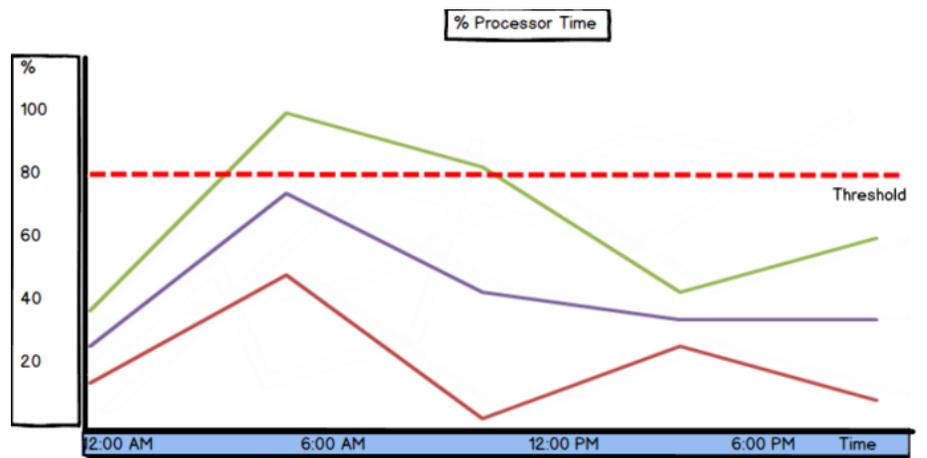


Figura 11 – Uso porcentual de recursos y su umbral determinado

2.6 Conocimiento asistido

Lograr conocimiento usando los datos históricos de rendimiento (performance) de los servicios informáticos, para explotarlos y obtener información útil de sus parámetros normales de operación.

El conocimiento asistido, es aquel que se puede lograr con la asistencia de una herramienta técnica, que permita interpretar y contrastar datos históricos, para provocar un conocimiento inmediato en las personas.

Este no pretende inferir aspectos de comportamiento con respecto a los componentes medidos, sin embargo se considera una poderosa herramienta para asistir el conocimiento que se va creando con respecto al uso de un servicio informático.

La modalidad de conocimiento asistido es hoy por hoy la herramienta más poderosa en el uso diario de tecnología, pues es tangible y realista en torno a las decisiones de tipo compleja que pueda tomar una persona.



Figura 12 – Ícono de conocimiento asistido

2.6.1 Conocimiento artificial

El conocimiento artificial, es aquel en que el análisis de un comportamiento complejo (toma en cuenta varios tópicos relacionados respecto a un suceso) es capaz de deducir artificialmente y por sí mismo los parámetros considerados normales.

El conocimiento ha sido estudiado desde muchos puntos de vista por un destacado biólogo chileno que es Humberto Maturana.

Ha sido realmente importante la lectura de este tema, usando en el presente trabajo el texto de “El árbol del conocimiento” del cual Humberto Maturana es coautor conjuntamente con otro destacado biólogo nacional que es Francisco Varela.

Han sido importantes porque permite entender el proceso que lleva a la mente humana a obtener un conocimiento a través de la experiencia y luego, como este conocimiento servirá para tomar decisiones en el presente.

De esta forma, se puede obtener un grado de conocimiento respecto al comportamiento de algo, a través del estudio de su comportamiento y desempeño en un ámbito que lo relaciona.

(Humberto Maturana, 1984)

*La contribución de Maturana a esta nueva proposición epistemológica es fundamental. Él es, uno de los primeros científicos de la biología que propusieron que el **conocer** es un fenómeno biológico que puede solamente ser estudiado y conocido como tal, y que ha desarrollado una completa teoría biológica consistente con esta mirada. Además, él propone que la misma vida debe ser entendida como un proceso de conocimiento, en la realización del vivir en congruencia con el medio. El trabajo de Maturana puede ser, por*

lo tanto, caracterizado como un sistema explicativo ontológico unitario de la vida y de la experiencia humana. Es ontológico porque visualiza a la experiencia humana desde un punto de vista situado dentro de las condiciones de constitución de lo humano y no desde una posición externa, y es explicativo porque propone una mirada de la dinámica de relaciones que genera los fenómenos del conocimiento.

En la medida que su entendimiento de los sistemas biológicos va emergiendo, el enfoque de Maturana nos lleva a reflexionar sobre las condiciones que nos permiten el explicar todo lo que ocurre en la vida como fenómeno del vivir..... En el pensamiento de Maturana, la mente es un fenómeno que pertenece a la dinámica relacional del organismo. En su mirada, la mente, como un fenómeno relacional, surge en la relación entre organismos y el medio de la misma manera que el caminar surge desde un movimiento de las piernas en relación con el suelo o como un desplazamiento del cuerpo.

2.6.2 Conocimiento experto

El conocimiento experto es aquel que ha sido obtenido a través de la experiencia, por lo tanto está ligada a la interpretación y análisis del comportamiento de algo, en un período de tiempo pasado y que puede ser contrastado con datos del presente, para determinar normalidades o el comportamiento esperado. (Howard Gardner, 1987)

Este conocimiento, para poder ser determinado como un conocimiento de tipo experto, debe cubrir algún área de interés para el ser humano, pues de la interacción con él, nacen situaciones o comportamientos dignos de analizar y por supuesto, de utilidad para grupos humanos.

2.6.3 Madurez del conocimiento a través de la ciencia cognitiva

La madurez del conocimiento es imprescindible en el entendimiento humano respecto de los medios y herramientas que este ocupa en sus labores cotidianas.

Es un proceso lento, que requiere una interpretación del actuar a través del tiempo y que usualmente se va creando con la supervisión y control cercano del comportamiento de un componente.

Hoy en día, como ya hemos explicado, el conocimiento se logra poniendo foco en componentes determinados o bien a través del seguimiento de casos de ruptura de los umbrales de los niveles de operación, que obligan a la intervención de grupos técnicos, para determinar o descartar un problema.

El uso de esta información y su procesamiento analítico, son procesos que se generan en las personas, o más bien dicho, en la mente de las personas, pero que no quedan graficados ni registrados en herramienta alguna.

Así entonces, en el período siguiente, si el comportamiento es similar (el evento o incidente ocurre nuevamente), el conocimiento adquirido sólo puede ser aplicado si aún es analizada por la misma persona o el mismo grupo técnico que participó en el período anterior, por lo tanto genera 2 problemas conocidos:

1. El análisis es aproximado, sin la exactitud requerida para los ámbitos empresariales.
2. Dependerá de que las personas continúen en la organización encargada de la supervisión.

El trabajo planteado puede lograr un conocimiento adquirido de forma artificial y que pretende mostrar la información y deducciones de una forma instantánea y simple, dejando a los grupos técnicos la toma de decisiones.

Lo que se espera es que un analista técnico responsable del servicio, pueda confirmar o descartar un problema, con información oportuna y de forma eficiente, aprovechando los datos explícitos presentados por la herramienta técnica.

III. DESARROLLO DEL TRABAJO

3.1 Metodología

El tipo de metodología que se usará en el presente trabajo, será empírico y estadístico, abordando principalmente estudios del comportamiento de servicios informáticos e intentar obtener un medio de interpretación y utilidad respecto del conocimiento que se obtiene de ellos.

Se utilizará técnicas de análisis de dispersión estadística métodos técnicos, que permitan identificar rangos normales de operación en cada período de tiempo y hacer una comparación de estos con la operación actual.

De esta forma, se plantea el objetivo de lograr determinar cuáles serán los límites de operación respecto a un período de tiempo similar en el pasado.

Diferentes métodos de cálculo matemático y estadístico, permitirán ensayar la utilidad que estos datos históricos puedan entregarnos, respecto a los niveles de normalidad de operación.

3.2 Planteamiento del problema

El problema planteado, es específicamente lograr determinar si el comportamiento de un componente de infraestructura, perteneciente a un servicio informático, tiene un comportamiento semi-predecible y posteriormente determinar los umbrales de operación que se puedan comparar con las métricas en el presente.

De acuerdo a esto, el planteamiento está enfocado a trabajar 3 problemas principales:

1. Determinar cuál es la mejor función estadística, para determinar los márgenes de operación esperados en el presente, de acuerdo a la información analizada del pasado, considerando que los sistemas que cumplen un patrón permanente deben tener un rango de operación más acotado y los sistemas que son más dispersos deben tener un rango mayor. Lo anterior es muy importante para no estar generando información errónea o falsos avisos de problemas en los componentes medidos.
2. Determinar los rangos y límites de lo que se considera una operación normal para el componente medido. A través de la información anterior, generar los límites de operación calculados y contrastarlos con los datos de consumo actuales en el componente medido (servidor).
3. Mostrar la información obtenida de una forma gráfica simple, para poder aportar en las decisiones de grupos técnicos y para poder ejercer un control inteligente sobre el servicio, más allá del simple control sobre los rangos de carga soportados por el componente. Se espera que esta información pueda ser desplegada en tiempo real, con una mirada instantánea de la operación y sus rangos de funcionamiento.

3.3 Posibles soluciones

La toma de muestras de métricas de consumo de un componente informático, es una práctica habitual en las áreas de informática de talla empresarial, por lo tanto este trabajo no se enfocará en métodos de medición, pero si aprovecharemos estos números para el estudio y análisis, para lograr exponerlos en una representación gráfica del antes y el ahora.

Problema 1

Como determinar si la medición de un componente cumple algún patrón de comportamiento.

Solución: Estudiar el comportamiento histórico en segmentos horarios similares entre un horario del pasado y el presente.

Así por ejemplo, como se presenta el siguiente cuadro, si el día actual es el Lunes de la primera semana de marzo del 2015, se comparará con el primer Lunes de la misma semana del año anterior.

2014					
Marzo					
semana 1					
Lunes			Martes		
zona 1	zona 2	zona 3	zona 1	zona 2	zona 3

2015					
Marzo					
semana 1					
Lunes			Martes		
zona 1	zona 2	zona 3	zona 1	zona 2	zona 3

Figura 13 – Zonas horarias de estudio

Los segmento horarios que usaremos para las comparaciones, son los usados habitualmente por Capacity Management (que es capacity management, software, métricas, etc...)

ZONA 1: Lunes a Domingo, desde la medianoche hasta las 08am

ZONA 2: Lunes a Domingo, desde las 8am hasta las 6pm

ZONA 3: Lunes a Domingo, desde las 6pm hasta la medianoche

Problema 2

Determinar los rangos y límites de lo que se considera una operación normal para el componente medido.

Solución: Para determinar esto, se necesita estudiar las medidas de dispersión que han tenido las métricas históricas, contrastando períodos de operación similares. Para el caso de nuestro trabajo, las comparaciones serán días de la semana similares, para meses similares y semanas similares (de acuerdo al ejemplo planteado en el punto 1). La tesis matemática será desarrollada en el punto siguiente.

Problema 3

Cómo mostrar la información obtenida de una forma gráfica simple.

Solución: La idea en este problema es identificar la información que sea sensible y útil para poder expresar de una forma simple el estado de los niveles de operación actuales.

Se plantea el uso del siguiente esquema gráfico, como una representación adecuada para los datos.

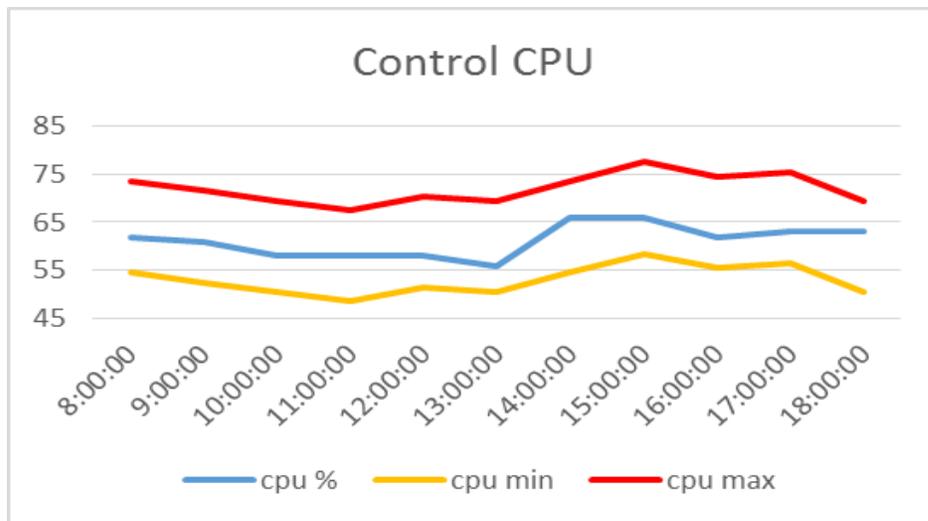


Figura 14 – Monitoreo de uso de CPU con límite inferior y superior

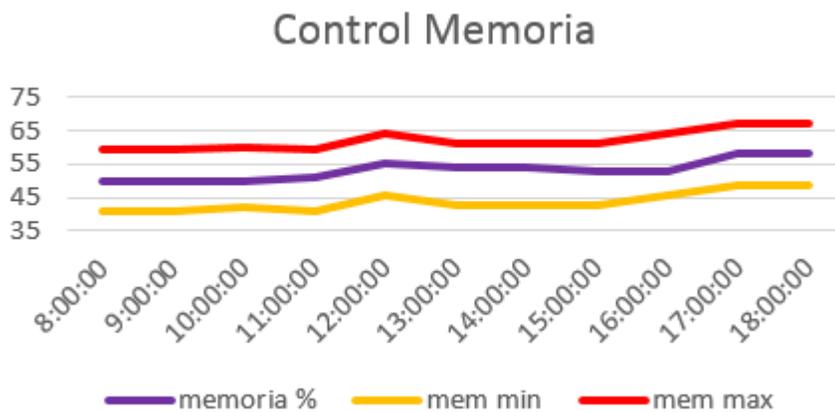


Figura 15 – Monitoreo de uso de Memoria con límites inferior y superior

En el gráfico, se puede apreciar las líneas amarilla y roja, representando los límites inferior y superior respectivamente, calculados usando los datos históricos.

Entre ellos se desarrolla la línea azul, con la representación de la operación actual instantánea del consumo de CPU (siendo actualmente las 18:00 hrs - el ahora en el ejemplo).

En el caso del ejemplo, la línea azul correspondiente a la operación actual, se comporta de acuerdo a los márgenes esperados, de acuerdo al cálculo de los datos históricos.

En este ejemplo, el cuadro de datos históricos y mediciones actuales, sería el correspondiente al siguiente cuadro.

zona horaria	16-10-2014		15-10-2015 (now)		rangos calculados cpu		rangos calculados memoria	
	cpu %	memoria %	cpu %	memoria %	cpu min	cpu max	mem min	mem max
zona 2: 08-18 hrs								
8:00:00	64	50	62	50	60,92	67,08	46,97	53,03
9:00:00	62	50	61	50	58,92	65,08	46,97	53,03
10:00:00	60	51	58	50	56,92	63,08	47,97	54,03
11:00:00	58	50	52	51	54,92	61,08	46,97	53,03
12:00:00	61	55	54	55	57,92	64,08	51,97	58,03
13:00:00	60	52	56	54	56,92	63,08	48,97	55,03
14:00:00	64	52	60	54	60,92	67,08	48,97	55,03
15:00:00	68	52	65	53	64,92	71,08	48,97	55,03
16:00:00	65	55	62	53	61,92	68,08	51,97	58,03
17:00:00	66	58	63	58	62,92	69,08	54,97	61,03
18:00:00	60	58	63	58	56,92	63,08	54,97	61,03
average	62,55	53,00						
Desv. Estándar	3,08	3,03						

Figura 16 – Rangos calculados (comparación histórica y actual)

3.4 Análisis matemático

Como se muestra en el cuadro, el presente trabajo propone como mejor medida de dispersión para el análisis de los límites, el uso de la varianza.

La **varianza** es una medida estadística que mide la dispersión de los valores respecto a un valor central (media), es decir, es el cuadrado de las desviaciones.

$$S_X^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

Propiedades [\[editar \]](#)

- La varianza es siempre positiva o 0: $V_X^2 \geq 0$
- Si a los datos de la distribución les sumamos una cantidad constante la varianza no se modifica.

Figura 17 – Cálculo de la varianza

Sin embargo, la representación de unidades cuadráticas no es posible en los mismos rangos de las mediciones, pero este problema queda resuelto con el cálculo de la Desviación Estándar.

La Desviación Estándar es la raíz cuadrada de la varianza de la distribución de probabilidad discreta:

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Figura 18 – Cálculo de la Desviación Estándar

Lo que se propone entonces, es calcular la Desviación estándar de los datos históricos comprendidos en un período de tiempo anterior comparable (ejemplo: zona 2, el mismo primer lunes, de la primera semana, en el mismo mes) y a la medición de la misma hora restar la desviación estándar para determinar el límite inferior y sumar la desviación para determinar el límite superior de normalidad.

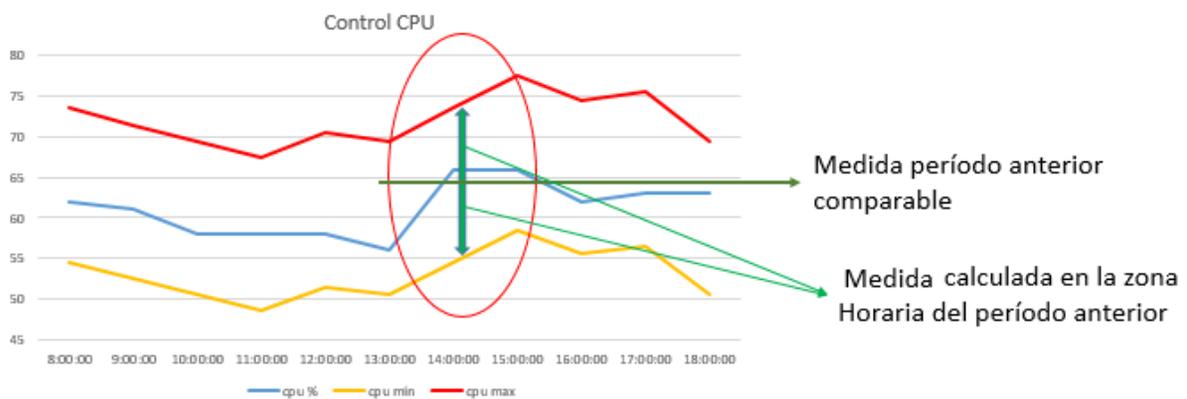


Figura 19 – Explicación de rango medido y sus umbrales (mínimo, máximo)

Así por ejemplo, si la variación de niveles de carga ha sido pequeña en períodos anteriores, los límites graficados serán más acotados, pues se trata de un sistema de comportamiento regular y por lo tanto más predecible.

Si por el contrario, los niveles de carga calculados del histórico han tenido una mayor variación, los límites se hacen más extensos, dando mayor espacio de operación antes de notificar un problema.

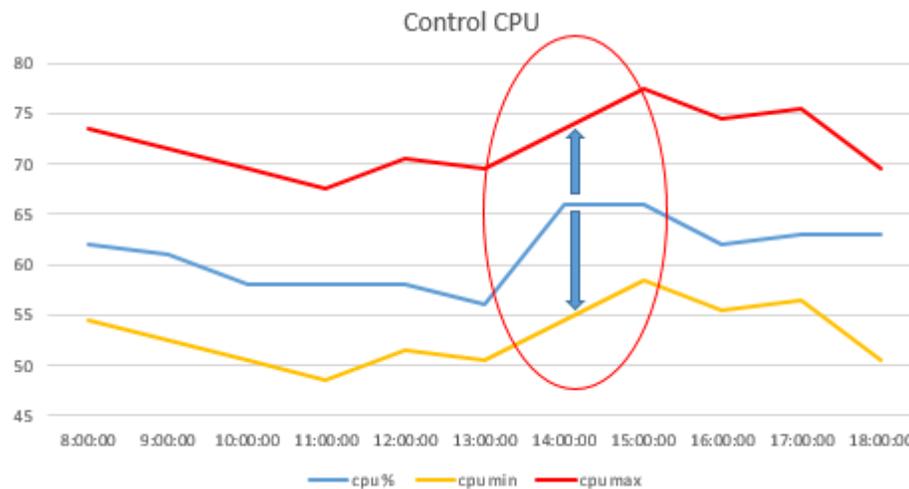


Figura 20 – Extensión de umbrales de acuerdo a información histórica

3.5 Conocimiento asistido

La idea central del presente trabajo y sus cálculos derivados, es aprovechar los datos históricos de consumo respecto a un servicio informático, para otorgar conocimiento a los grupos humanos, facilitando así el control y la toma de decisiones.

En términos generales, el conocimiento de normalidad de operación de estos servicios, se logra con la experiencia humana de control sobre las plataformas, pero generalmente no se aprovecha los datos históricos.

Así por ejemplo, cuando un grupo humano pretende determinar si los niveles actuales de operación son normales, tendrá que hacer un cálculo manual, que en ningún caso le prestará información en línea, además de que no lo podrá resolver en el período de tiempo que desea medir.

Adicionalmente, no tendrá la gráfica de una forma instantánea y fácil de leer, para poder confirmar o descartar un problema.

A través del presente trabajo, exponemos una forma de obtener y graficar la información requerida en tiempo real, lo que permitirá hacer un seguimiento de las condiciones del sistema, incluso si no existe un conocimiento anterior del mismo.

Así, se espera que cualquier analista pueda acceder a los datos de conocimiento, para determinar inmediatamente si el sistema está otorgando prestaciones normales o ante una situación excepcional.

Por ejemplo, en el siguiente gráfico, se expone una situación en que claramente existe un problema (deducción desde los límites de operación presentados) y en el cual es altamente conveniente la revisión de un especialista.

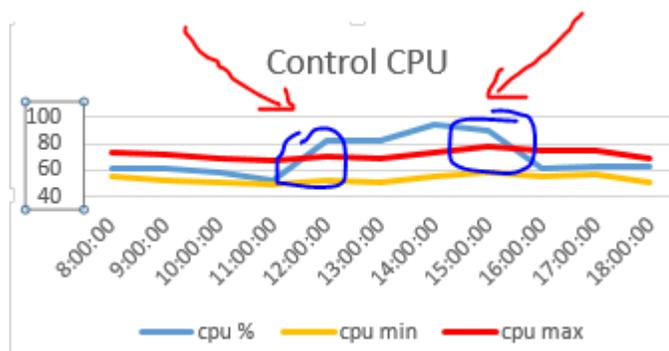


Figura 21 – Niveles de operación (CPU) queiebran el umbral superior

Se deduce que en el período anterior, la operación en el mismo segmento horario (**zona 2** 08:00 a 18:00 hrs) era bastante estable y en el período actual tuvo 2 queiebres del límite superior esperado para el servicio (indicados en azul).

3.6 Módulo de medición

El módulo de medición es el script que extraerá los datos históricos, hará el cálculo de los parámetros y rangos requeridos, para que sean aprovechados por la herramienta gráfica y presentados en un panel a los grupos técnicos que lo requieran.

Como ejemplo en el presente trabajo, el desarrollo será un script usando lenguaje de Shell en un sistema Linux.

Se dejará como un hallazgo, la oportunidad de desarrollar el script para su uso en varios tipos de sistema operativo y así cubrir la mayor cantidad de sistemas posibles.

Esto no se considera como el foco del estudio, por lo que podrían ser abordados en trabajos complementarios.

3.6.1 Módulo de medición local

El primer script que fue necesario construir, es el de mediciones locales (en cada uno de los servidores), que serán invocados remotamente (por el servidor de monitoreo central).

Estos scripts nos proveerán 2 datos: El porcentaje de uso del tópico específico (memoria o CPU) y el resultado de la medición (estado de la lectura).

A continuación se detallan ambos, elaborados en lenguaje de Shell. Tienen una utilidad general de uso para los sistemas basados en Nagios, es decir, mantienen un formato estándar, que le permite funcionar con cualquiera de los sistemas basados en NRPE.

Para nuestro caso, sólo ocuparemos (recortando el string de respuesta) el porcentaje de uso del tópico particular (CPU/MEM).

```
#!/bin/bash
#
# Nagios CPU Plugin
#
# Description: Check the cpu status
# Author      : Jorge Alvear
# Version     : 1.0
#

prog=`basename $0`

if [ $# -lt 2 ]; then
    echo "Usage: $0 <warning> <critical>"
    exit 2
fi

# CPU threshold values
CPU_WARNING=$1
CPU_CRITICAL=$2

if (( $CPU_WARNING >= $CPU_CRITICAL ))
then
    echo "<WARNING> value must be less than the <CRITICAL> value"
    exit 5
fi

OUTPUT_USE=`vmstat 1 5|egrep -v "Time|kthr"|awk '{print $13}'|tail -1`
OUTPUT_IDLE=`vmstat 1 5|egrep -v "Time|kthr"|awk '{print $15}'|tail -1`

if (( $OUTPUT_USE < $CPU_WARNING ))
then
    echo "OK - CPU Utilization is OK $OUTPUT_IDLE% idle | CPU=$OUTPUT_USE"
    exit 0
fi
```

```
elif (( $OUTPUT_USE >= $CPU_CRITICAL ))
then
    echo "CRITICAL - CPU Utilization is TOO HIGH $OUTPUT_IDLE% idle |
CPU=$OUTPUT_USE"
    exit 2
elif (( $OUTPUT_USE >= $CPU_WARNING ))
then
    echo "WARNING - CPU Utilization is HIGH $OUTPUT_IDLE% idle |
CPU=$OUTPUT_USE"
    exit 1
else
A
    echo "CPU STATUS UNKNOWN"
    exit 3
fi
```

Caso script para medición de memoria.

```
#!/bin/bash
#
# Nagios MEM Plugin
#
# Description: Check the cpu status
# Author      : Jorge Alvear
# Version     : 1.0
#
case $1 in
    --help | -h )
        echo "Plug In for check % of memory use "
        echo "Usage: check_mem [warning] [crit]"
        echo " [warning] and [crit] as int"
        echo " Example: check_mem 80 90"
        exit 3
        ;;
    * )
```

```
;;
esac

MEM_WARNING=$1
MEM_CRITICAL=$2

if [ ! "$1" -o ! "$2" ]; then
    echo "Usage: check_mem [warning] [critical]"
    echo " [warning] and [critical] as int"
    echo " Example: check_mem 80 90"
    echo "Unknown: Options missing: using default (warning=80,
critical=90)"
    MEM_WARNING=`echo $((80))`
    MEM_CRITICAL=`echo $((90))`
fi

OUTPUT=$(free |head -2 | tail -1))

MEM_TOTAL=${OUTPUT[1]}
MEM_USED=${OUTPUT[2]}
MEM_FREE=${OUTPUT[3]}

MEM_USE=$((($MEM_USED * 100) / $MEM_TOTAL))

if (( $MEM_WARNING >= $MEM_CRITICAL ))
then
    echo "<WARNING> value must be less than the <CRITICAL> value"
    exit 5
fi

if (( $MEM_USE < $MEM_WARNING ))
then
    echo "OK - Use=$MEM_USE% | Mem_Use=$MEM_USE"
    exit 0
elif (( $MEM_USE >= $MEM_CRITICAL ))
then
```

```
    echo "CRITICAL - Memory use is Critical; Use=$MEM_USE%| Mem_Use=$MEM_USE"
    exit 2
elif (( $MEM_USE >= $MEM_WARNING ))
then
    echo "WARNING - Memory use is High; Use=$MEM_USE%| Mem_Use=$MEM_USE"
    exit 1
else
    echo "UNKNOWN"
    exit 3
fi
```

3.6.2 Módulo de medición remota

Para tomar estas mediciones en forma centralizada (desde el servidor de monitoreo), se construyó un script (check_icontrol) que permitirá ejecutar los scripts de CPU y Memoria en forma remota, que requiera sólo un usuario y clave.

Esto es muy importante al momento de trabajar en entornos productivos, pues en estos ambientes se dificulta la instalación de agentes o software, los que requieren una serie de aprobaciones y pruebas, que complican su uso.

La solución para este problema está basada en la utilidad java RemoteCommand, que permite justamente esta funcionalidad, ejecutar comandos de forma remota, requiriendo sólo un usuario y clave válidos.

A continuación el script que ejecuta los scripts locales, extrayendo las métricas para la comparación con los umbrales establecidos.

```
#!/bin/bash
#
# Nagios check_icontrol
#
# Description: check_control execute a command using remote_Command.jar
# Jar is located in poller /var/spool/nagios/plugins
# host/user/pass must be defined in /var/spool/nagios/hostnames.csv
# MEASURE compare metrics with calculated thresholds
# Author      : Jorge Alvear
# Version     : 1.0
#

prog=`basename $0`

if [ $# -ne 3 ]; then
    echo "Error: usage: $0 <TOPIC> <HOSTNAME> <COMMAND>"
    exit 2
fi

#Parameters
TOPIC=$1
HOSTNAME=$2
COMMAND=$3

#IC_PATH
IC_PATH="/etc/icontrol"

OUTPUT=`/opt/jdk1.6.0_45/bin/java                                -jar
/var/spool/nagios/plugins/remote_Command.jar $HOSTNAME "$COMMAND"`

MEASURE=${OUTPUT##*=}

IC_OUTPUT=`/opt/jdk1.6.0_45/bin/java                            -cp          $IC_PATH:$IC_PATH/lib
IntelligentControl $IC_PATH $TOPIC $HOSTNAME`
```

```
THRESHOLDS=(${IC_OUTPUT})

UPPER_LIMIT=${THRESHOLDS[1]}
LOWER_LIMIT=${THRESHOLDS[2]}

if [[ $MEASURE -gt $LOWER_LIMIT ]] && [[ $MEASURE -lt $UPPER_LIMIT ]];
then
    echo "OK - $TOPIC Utilization is IN RANGE |"
    $TOPIC=$MEASURE,UPPER=$UPPER_LIMIT,LOWER=$LOWER_LIMIT"
    exit 0
else
    echo "WARNING - $TOPIC=$MEASURE% Utilization is OUT OF RANGE |"
    $TOPIC=$MEASURE,UPPER=$UPPER_LIMIT,LOWER=$LOWER_LIMIT"
    exit 1
fi
```

Este script, utilizará la clase java IntelligentControl, que es en definitiva la que usa los datos y genera los cálculos.

El script de ejecución remota, requiere de la lectura de un XML con parámetros básicos de funcionamiento.

Toda la implementación presentada, está orientada a otorgar la máxima portabilidad entre sistemas y ambientes.

options.xml

```
<options-hputils>
  <options
    calc-type="thresholds"
    dir-data="/data/"
    time-compare="year"
    float-point="."
    float-precision="3"
  >
</options>
</options-hputils>
```

3.7 *Módulo de Control Inteligente*

El módulo de control inteligente, es quien lleva el procesamiento de la información obtenida, ejecutando el análisis histórico obtenido antes y determinando los umbrales de funcionamiento normal.

Además manejará los estados de consulta y entregará un string que sea válido para los sistemas de monitoreo basados en NRPE, de modo que la información pueda ser graficada de manera simple.

Una vez más, pensando en la portabilidad y utilidad del desarrollo, se ha elaborado en tecnología JAVA.

```
import java.io.*;
import java.text.*;
import java.util.*;

public class IntelligentControl{

private static BufferedReader reader = null;
    private static XMLReader xmlReader;

    private static SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-
YYYY");

    private static boolean verbose = true;

    // get the supported ids for GMT-08:00 (Pacific Standard Time)
    private static String[] ids = TimeZone.getAvailableIDs(-3 * 60 * 60 *
1000);

    // if no ids were returned, something is wrong. get out.
    // if (ids.length == 0) System.exit(0);

    // create a Pacific Standard Time time zone
    private static SimpleTimeZone pdt = new SimpleTimeZone(-3 * 60 * 60 *
1000, ids[0]);
    //SimpleDateFormat sdf = null;

    // set up rules for Daylight Saving Time
    //pdt.setStartRule(Calendar.APRIL, 1, Calendar.SUNDAY, 2 * 60 * 60 *
1000);
    //pdt.setEndRule(Calendar.OCTOBER, -1, Calendar.SUNDAY, 2 * 60 * 60 *
1000);

    //private static String NEW_FORMAT = "dd-MM-yyyy HH:mm";
    private static String NEW_FORMAT = "dd-MMM-yyyy";
    private static SimpleDateFormat sdf = new SimpleDateFormat(NEW_FORMAT);
    private static Calendar calendar = new GregorianCalendar(pdt);
```

```
private static Date nowDate = new Date();

private static String hostname = "";
private static String topic = "";

private static String default_path = "/etc/icontrol/";

// Main section
public static void main(String[] args) {

    calendar.setMinimalDaysInFirstWeek(1);
    calendar.setFirstDayOfWeek(Calendar.MONDAY);

    if (!(args.length > 0) && (args != null) && (args[0] != null)){
        //System.out.println("Error: Problemas con los <parametros>");
        System.out.println("use: IntelligentControl <HOSTNAME> <TOPIC>");
        System.exit(1);
    }

    //System.out.println("Procesing file='" + file_name + "'");

    default_path = args[0];

    String file_name = default_path + "/options.xml";

    hostname = args[2].toLowerCase();

    if (hostname.contains(".")){
        hostname = hostname.substring(0,hostname.indexOf("."));
    }

    topic = args[1].toUpperCase();

    xmlReader = new XMLReader(file_name, false);
```

```
        if (xmlReader.getAttribute("options", "calc-  
type").equals("thresholds")) processData();  
    }  
  
    private static void processData(){  
        try {  
            String dir_data = xmlReader.getAttribute("options", "dir-  
data");  
            String float_point = xmlReader.getAttribute("options",  
"float-point");  
  
            String[] args;  
  
            // leera los datos y almacenará  
  
            int lower = 0;  
            int upper = 0;  
            int measure = 0;  
            int zone = 0;  
  
            int position= 0;  
  
            int stddev= 0;  
  
            int day_inFile = 0;  
            int day_last = 0;  
  
            String data_file = "";  
            String line = "";  
  
            Date date = null;  
  
            calendar.setTime(getPastDate());  
            day_last = calendar.get(Calendar.DAY_OF_MONTH);
```

```
//System.out.println("PastDate=" + getPastDate());

    data_file = "data_" + hostname + "_" + topic + "_" +
calendar.get(Calendar.MONTH) +
        calendar.get(Calendar.YEAR) + ".csv";

//System.out.println("reading file='" + default_path + dir_data +
data_file + "'");
    reader = new BufferedReader(new
FileReader(default_path + dir_data + data_file));

    while ((line = reader.readLine()) != null) {

        if (!line.startsWith("#")){

            args = line.split(";");
            date = sdf.parse(args[0]);

            calendar.setTime(date);
            day_inFile = calendar.get(Calendar.DAY_OF_MONTH);

            if (day_inFile == day_last) {

                zone = getZone(nowDate);

                switch (zone) {
                    case 1: position = 0;
                        break;
                    case 2: position = 2;
                        break;
                    case 3: position = 4;
                        break;
                }
            }
        }
    }
}
```

```
                measure = (int)
Math.round(Float.parseFloat(args[1 + position]));
                stddev = (int)
Math.round(Float.parseFloat(args[2 + position]));

                lower = measure - stddev;
                upper = measure + stddev;

                if (upper > 99) upper = 99;
                if (lower < 1) lower = 1;

            }
        }
    }

    reader.close();

    System.out.println("OK " + upper + " " + lower);

} catch (Exception e) {
    System.out.println("ERROR;0;0");
    e.printStackTrace();
}
}

private static Date getPastDate() {

    Date lastDate = null;

    try{
        // create a GregorianCalendar with the Pacific Daylight time zone
        // and the current date and time
        Calendar calendar = new GregorianCalendar(pdt);
        Calendar calendar_last = new GregorianCalendar(pdt);

        calendar.setMinimalDaysInFirstWeek(1);
```

```
calendar_last.setMinimalDaysInFirstWeek(1);

calendar.setFirstDayOfWeek(Calendar.MONDAY);
calendar_last.setFirstDayOfWeek(Calendar.MONDAY);

// setear la fecha
// nowDate = sdf.parse("31-10-2015 16:00");
lastDate = nowDate;

// System.out.println("rev date=" + lastDate);
calendar.setTime(nowDate);

        calendar_last.set(GregorianCalendar.YEAR,
calendar.get(Calendar.YEAR));
        calendar_last.set(GregorianCalendar.MONTH,
calendar.get(Calendar.MONTH));

// calendar_last = calendar;
if (xmlReader.getAttribute("options", "time-compare").equals("year")){
        calendar_last.set(GregorianCalendar.YEAR,
calendar.get(Calendar.YEAR) - 1);
    } else {
        calendar_last.set(GregorianCalendar.MONTH,
calendar.get(Calendar.MONTH) - 1);
    }

        calendar_last.set(GregorianCalendar.WEEK_OF_MONTH,
calendar.get(Calendar.WEEK_OF_MONTH));
        calendar_last.set(GregorianCalendar.DAY_OF_WEEK,
calendar.get(Calendar.DAY_OF_WEEK));

        calendar_last.set(GregorianCalendar.HOUR_OF_DAY,
calendar.get(Calendar.HOUR_OF_DAY));

/*
```

```
        calendar_last.set(GregorianCalendar.MINUTE,
calendar.get(Calendar.MINUTE));
        calendar_last.set(GregorianCalendar.SECOND,
calendar.get(Calendar.SECOND));
        */

        lastDate = calendar_last.getTime();

    } catch (Exception e) {
        e.printStackTrace();
    }

    return lastDate;
}

private static int getZone(Date date) {

    int hour = 0;

    try{
        calendar.setTime(date);
        hour = calendar.get(Calendar.HOUR_OF_DAY);

    } catch (Exception e) {
        e.printStackTrace();
    }

    if ((hour > -1) && (hour < 9)) {
        return 1;
    } else {
        if ((hour > 8) && (hour < 17)) {
            return 2;
        } else {
            return 3;
        }
    }
}
```

```
}  
}
```

El script tuvo varios desafíos interesantes en su construcción:

La búsqueda del día comparable, se logró a través de la implementación que JAVA tiene para el calendario Gregoriano (GregorianCalendar). Esto permitió por ejemplo especificar un día, dentro de la semana, dentro de un mes. (por ejemplo: determinar cuál fue el segundo Lunes en Diciembre del 2014, para compararlo con el mismo día del presente año).

Otro desafío interesante, fue como comparar las fechas buscadas en contraste con las fechas de las fuentes de datos. Esto es posible con la implementación que java tiene de SimpleDateFormat, que permite ubicar patrones predeterminados y convertirlos a un formato de fecha y hora estándar.

Otro desafío interesante fue el cálculo de la desviación estándar, dentro de una zona horaria determinada en el día. El cálculo en línea era demasiado lento, debido a la cantidad de mediciones históricas que contienen diariamente los sistemas de monitoreo, por lo tanto la mejor solución fue calcularla anticipadamente y disponerla en la misma fuente de datos.

3.8 Representación gráfica de la información

La representación gráfica de la información, se logrará haciendo uso de un sistema simple y comúnmente utilizado para controlar los niveles de operación, que es Centreon.

Centreon permite graficas cualquier parámetro de consumo de un servicio informático y representarlo gráficamente en línea.

También tiene la particularidad de generar alarmas si es necesario, para nuestro caso sería útil, pues permite generar alarmas si los límites inferiores o superiores son excedidos. Esto será considerado como un hallazgo, sobre el cual se puede profundizar como una extensión del trabajo.

El sistema está basado en la tecnología NRPE y RRDTool, estándares que permiten obtener métricas y graficarlas de una forma fácil.

A continuación, algunas imágenes que explican el resultado de los componentes implementados para presentar la información.

Presentación estratégica (una vez que hay ingreso al sistema)

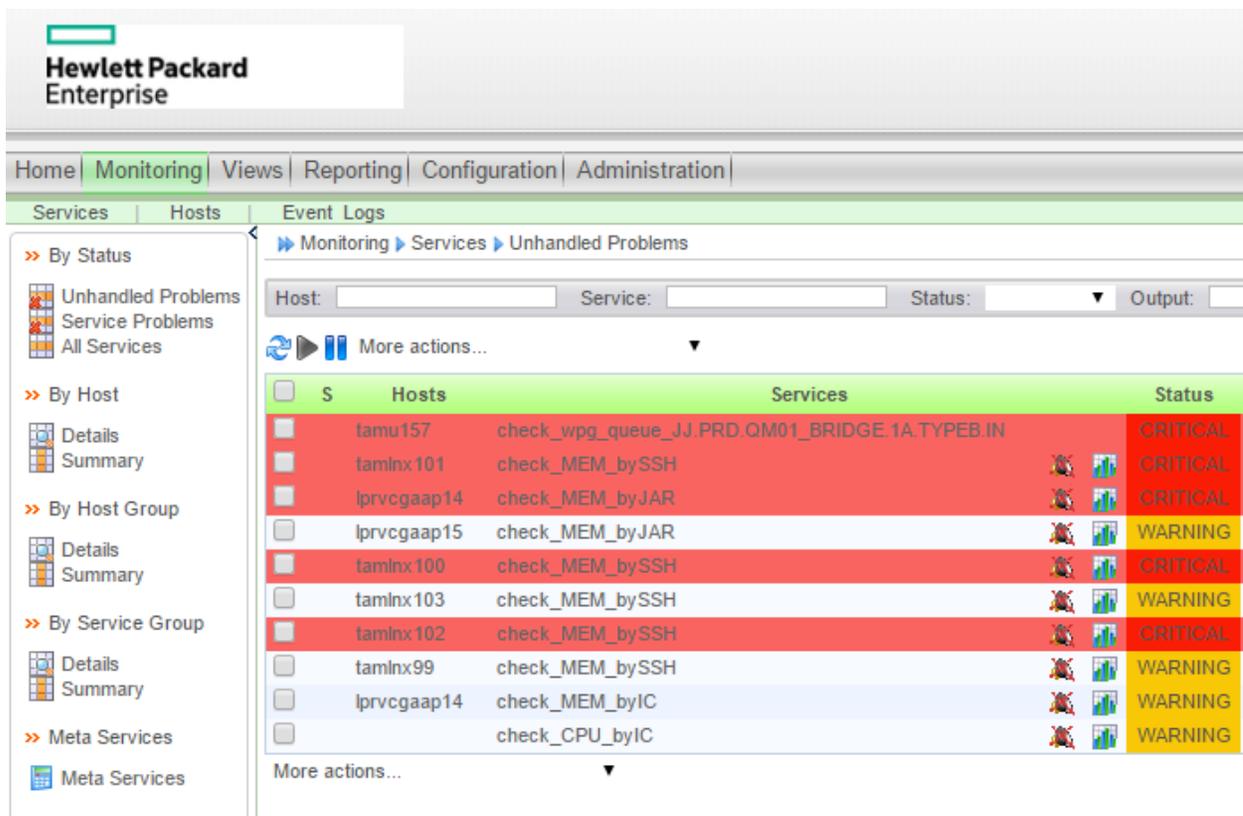


Figura 22 – Sistema de monitoreo Centreon

Presentación de medición simple de los componentes (check_CPU y check_MEM)

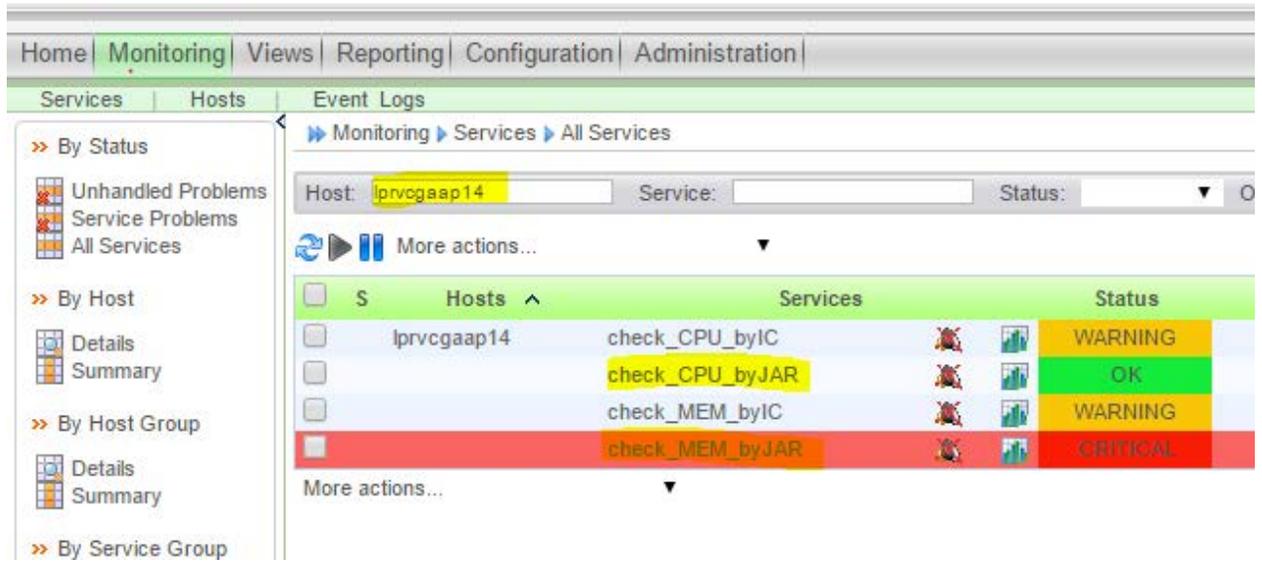


Figura 23 – Centreon Controlando un servidor de producción

Gráfica individual del tópico CPU, recolectada de forma central y expresada con sus estados.

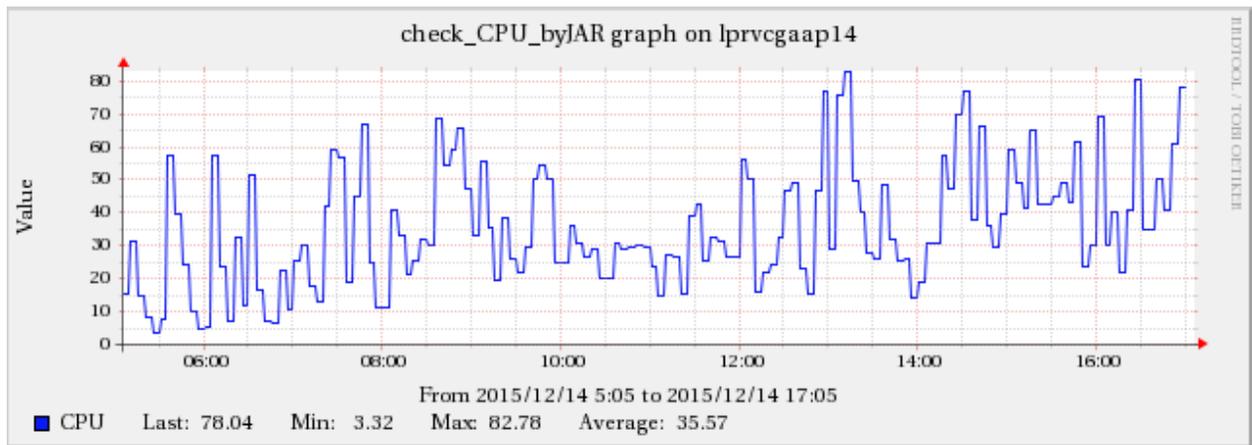


Figura 24 - Gráfica individual del tópico Memoria, recolectada de forma central y expresada con sus estados.

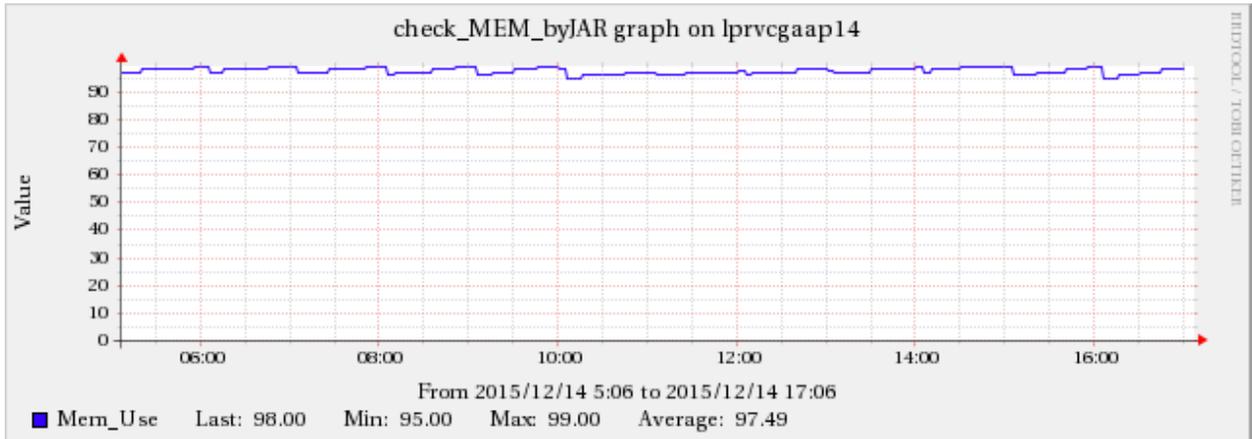


Figura 25 - Gráfica individual del tópico Memoria, recolectada de forma central y expresada con sus estados.

3.9 Gráfica de asistencia a la operación

La gráfica según lo descrito en el punto anterior, debe ser en línea y con información simple de los cálculos obtenidos.

Como se describe en el punto anterior, la gráfica de los cálculos obtenidos será presentada en línea en la herramienta centreon.

También se determinó un hallazgo importante, que es tener una tabla con el cálculo anticipadamente de los niveles de operación históricos y sus límites de operación, pues el cálculo en línea para una cantidad de servidores mayores, dará como resultado que se genere cierta lentitud en el sistema.

Las métricas de servicio se agruparon de la siguiente forma, incluyendo el cálculo de desviación estándar y la separación por zonas horarias.

#date	zona 1		zona 2		zona 3	
01-dec-14	56	21	57	19	51	26
02-dec-14	55	21	37	26	39	27
03-dec-14	49	23	34	24	34	20
04-dec-14	53	12	39	29	60	24
05-dec-14	59	32	32	27	72	15
06-dec-14	81	11	71	14	74	13
07-dec-14	84	8	84	9	87	6
08-dec-14	68	36	83	10	86	9
09-dec-14	86	13	65	16	73	16
10-dec-14	81	12	69	14	54	12
11-dec-14	66	10	64	14	72	13
12-dec-14	81	12	76	10	79	11
13-dec-14	81	11	74	11	76	13
14-dec-14	82	11	80	9	80	12
15-dec-14	84	10	83	13	83	12
16-dec-14	60	28	75	17	80	12

Figura 26 – Desviación estándar x segmento horario

Las métricas de servicio son presentadas en gráficos de fácil lectura, que permitan asistir la labor humana con una lectura rápida del estado actual.

Métricas para control inteligente de CPU

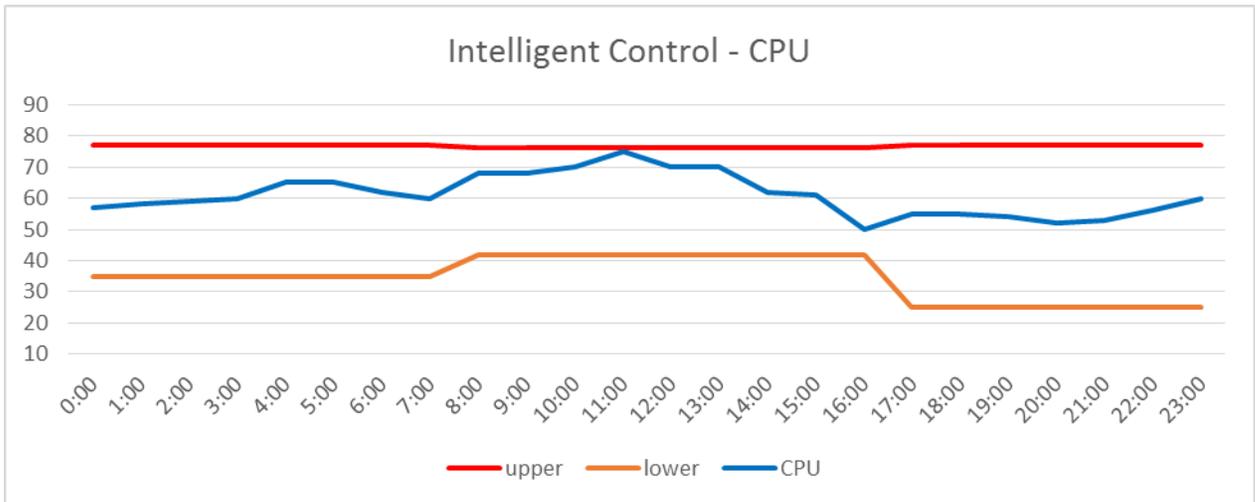


Figura 27 - Métricas para control inteligente de CPU

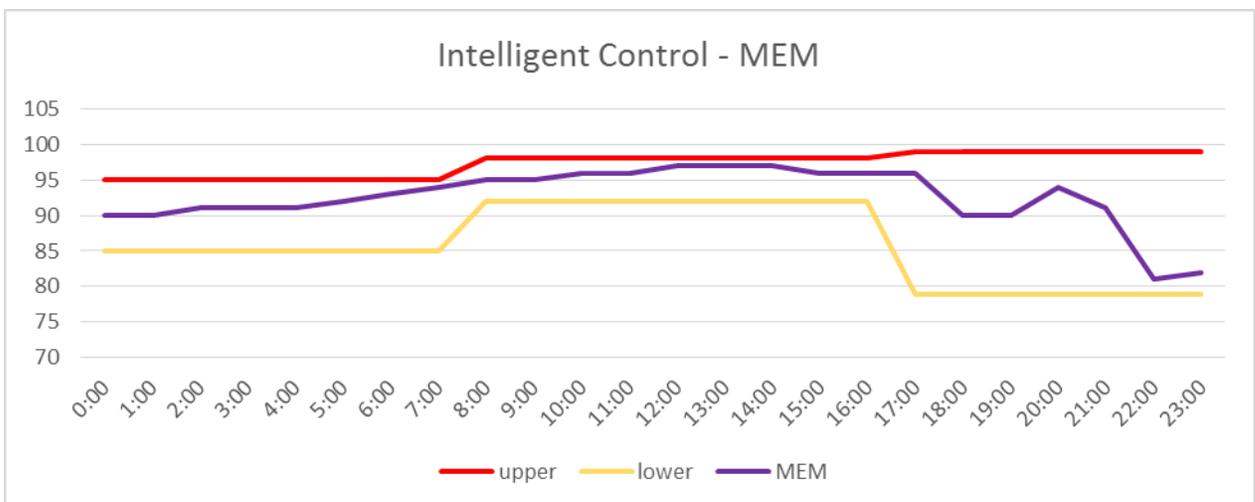


Figura 28 – Métricas para control inteligente de Memoria

IV. HALLAZGOS

4.1 *Contrastar más de un período de tiempo*

Poder analizar más de un período de tiempo pasado, para normalizar y acotar aún más el comportamiento esperado y sus límites de operación. De esta forma, aprovechando la experiencia de todo el tiempo pasado disponible, se podrá generar una asistencia de conocimiento mucho más certero.

4.2 *Generar alarmas de quiebre de umbrales*

Como un importante complemento del presente, se puede diseñar alarmas para los quiebres de umbrales determinados.

Esto permitirá la intervención de un especialista, si se considera necesaria la revisión o incluso determinar que existe algún problema o ataque informático.

4.3 *Agregar más zonas de comparación horaria*

Sería de mucha utilidad el poder comparar otro tipo de zonas horarias, como fin de semana, feriados, etc.

Esto permitiría una comparación aún más realista entre el tipo de días de diferente año y diferente mes.

4.4 Registrar los procesos top 10 en quiebres de umbrales

Al momento de que se quiebre un umbral de control, sería óptimo poder registrar los procesos en ejecución. Esto es vital en el análisis de problemas, pero la información no es registrada, por lo que a veces el análisis es sólo deductivo.

4.5 Normalizar períodos anuales

Cuando se cuenta con información extendida de varios años, puede ser muy útil para normalizar las mediciones y poder detectar lo que se conoce como crecimiento vegetativo, que es el aumento normal por ocupación de un determinado sistema.

V. CONCLUSIONES

El trabajo de investigación desarrollado, permite obtener las siguientes conclusiones:

5.1 *El cálculo matemático es posible*

El cálculo matemático esperado es posible a través del uso de la desviación estándar, lo que permite determinar un margen de operación normal. ¿Para qué?

5.2 *Es posible graficar la información de forma simple*

Es posible graficar de una forma simple estos límites, de manera que permitan a los especialistas una lectura rápida de la información, para determinar si requiere una revisión o se descarta/confirma un problema.

5.3 *Es posible graficar la información en tiempo real*

Es posible presentar la información en tiempo real, lo que permite un seguimiento exhaustivo de la operación y desempeño de un componente informático.

5.4 *Se comprueba la utilidad de la herramienta*

Se comprueba la utilidad que esta herramienta puede tener para los grupos especialistas y administradores de infraestructura.

Por ejemplo sería una herramienta imprescindible al momento de que un outsourcing tome el control de servicios tecnológicos, pues no tiene un conocimiento humano previo de la operación, que puede ser determinado y asistido usando los datos históricos recolectados por los sistemas de monitoreo.

5.5 *Excelente herramienta de apoyo*

El presente control de operación no reemplaza a los actuales sistemas de monitoreo basados en umbrales por capacidad, sino más bien se vuelve en un excelente complemento de ellos.

5.6 *Interpretación asistida en tiempo real*

La interpretación asistida en tiempo real es posible a través del uso de la data histórica, determinando un patrón de comportamiento o expandiendo los límites de operación si es que la variación de operación ha sido amplia.

GLOSARIO

Script

En informática, el script es un programa simple, que habitualmente tiene sólo una tarea determinada y se usa como parte de un programa o procesamiento más complejo.

Shell

Una Shell es una secuencia de ordenes o comandos que puede interpretar el sistema operativo Unix/Linux. Su utilidad radica en lograr una secuencia de ordenes simples para posteriormente ingresarla en una programación horaria.

Centreon

Herramienta de monitoreo de uso de componentes y aplicaciones que tiene la ventaja de muestras gráficas incorporadas. Es considerada una de las más maduras y se puede utilizar en régimen Open Source.

Capacity Management

Proceso de la especificación ITIL encargada del correcto aprovisionamiento de recursos y del estudio de las capacidades para el cumplimiento de operación.

Medidas de dispersión

Son las medidas que muestran la variabilidad de un conjunto de datos, indicando cuan alejados están de un punto medio.

Status healthy

Estado de desempeño de un componente informático considerado saludable o dentro de niveles de normalidad.

NRPE

Permite ejecutar plugins locales en equipos remotos de equipos Linux. El plugin `check_nrpe` es llamado por Nagios y hace las peticiones de los plugins hacia el equipo remoto.

Nagios

Sistema de monitoreo de redes ampliamente utilizado, de código abierto, que vigila los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado.

Centreon

Sistema de monitoreo y control que integra funcionalidades de alerta (como Nagios) y funcionalidades de gráfica (como Cacti).

Además proporciona una funcionalidad extendida para uso de múltiples satélites y diferentes tipos de sistema operativo.

Su potencia está en la gran cantidad de plugins y en una administración absolutamente gráfica.

JAVA

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para poder ser ejecutado en diferentes plataformas, usando un motor específico elaborado para tal efecto.

Su principal potencia es la portabilidad, lo que permite escribir el código una vez y ejecutar donde sea.

XML

Es un lenguaje de etiquetas desarrollado por el World Wide Web Consortium (W3C) utilizado para representar datos en forma estructurada.

En la actualidad se convierte en un estándar para transmitir datos a través de internet.

router

Es un equipo usado para conducir (enrutar) requerimientos de red a una red diferente de la local.

String

Cadena de caracteres, habitualmente almacenadas en memoria.

Outsourcing

Nomenclatura anglo para denominación de un servicio externalizado.

BIBLIOGRAFÍA

Internet

- www.conductahumana.com
- <http://www.ibmpressbooks.com/>
- <https://www.centreon.com/en/community/>
- <https://support.office.com/es-hn/article/Cursos-de-aprendizaje-videos-y-tutoriales-de-Excel-2013>

Libros

- ITIL Capacity Management – Larry Klosterboer, 2011.
- Capacity Management Process – Allison Kuzenko (HP Itil Process), 2015
- El árbol del conocimiento – Humberto Maturana y Francisco Varela, 1984
- Estadística y probabilidades – Aaron Estuardo Morales (UFRO) 2012
- Metodología de la investigación / Cómo escribir una tesis - Edelsys Hernández Meléndrez, 2006