

Oracle Business Transaction Management:

Eine Einführung

Arne Brüning
Oracle Deutschland B.V.
Hamburg

Schlüsselworte

Fusion Middleware, Enterprise Manager Cloud Control, Business Transaction Management, SOA

Einleitung

Oracle Business Transaction Management (OBTM) ist eine der eher weniger bekannten Komponenten des SOA Management Packs. Dieser Vortrag soll folgendes vermitteln:

- Überblick über BTM im allgemeinen: wer braucht das überhaupt, wer nicht?
- Einstieg in OBTM: wo fange ich an? Hinweise zu Installation und Setup.
- eine kleine live-Demo: wie benutzt man OBTM
- Abgrenzung zu verwandten/alternativen Techniken

Nach dem Vortrag sollte der Zuhörer ein Gefühl dafür bekommen haben, ob es sich für sie/ihn lohnt, sich weiter mit OBTM zu beschäftigen und wie man ggf. tiefer in diese Thematik einsteigen kann.

Warum BTM?

Verfahren zum System-Monitoring gibt es bereits viele. Warum wird Business Transaction Management als neue Kategorie also überhaupt benötigt? Muss man wirklich jeden neuen Hype der Hersteller mitmachen? Oder verbirgt sich hinter BTM eine wirklich interessante Kategorie von System Management Werkzeugen, die es Wert sind, weiter betrachtet zu werden?

Schaut man sich an, wie sich die Systemlandschaften in den letzten 10 Jahren entwickelt haben und gleicht man damit ab, wie die System Management Werkzeug-Anbieter darauf reagiert haben, kann man feststellen dass sich hier eine kleine Kluft ergibt.

Zurück in alten Client-Server Tagen war das Management noch vergleichsweise einfach. Auch auf das Aufkommen von Application Servern haben die Anbieter reagiert und können heute die ganze Palette von Storage, Hardware, Systemsoftware, Datenbank und Middleware verwalten. Mit der Einführung von lose gekoppelten Systemen, sei es nun über eine explizite SOA-Strategie, oder einfach aus den Projekten heraus, die SOA-Methoden benutzen: Um so mehr eine geschäftliche Transaktion über immer mehr lose gekoppelte Systeme erfolgt, um so schwieriger wird das Monitoring. Insbesondere wenn die Geschäftstransaktion als Prozess ausmodelliert wurde, und dieser in Abhängigkeit der Payload unterschiedliche Services nutzt, ist dies mit klassischem Monitoring schwer zu überwachen und im Fehlerfall muss sich jemand mit guter Systemkenntnis normalerweise durch verschiedene Systemüberwachungstools hangeln, bis die Ursache gefunden wurde.

Ausserdem sind mit klassischen Monitoring-Methoden fachliche Anfragen schwer zu beantworten. „Warum hängt der Auftrag #1234567?“ lässt sich im klassischen Monitoring nur mit viel Anwendungswissen beantworten.

Business Transaction Management Systeme gehen hier den umgekehrten Weg. Geschäftstransaktionen sind hier 1st Class Citizens, die über beliebig viele Systeme verteilt werden. Instanzen von Geschäftstranaktionen werden überwacht und die Frage nach dem Status eines Auftrags kann hier schnell beantwortet werden.

Dieser Vortrag soll Business Transaction Management am Beispiel des Oracle Products ‚Oracle Business Transaction Management‘ vorstellen.

Oracle BTM (OBTM)

Oracle Business Transaction Management (OBTM) kam durch die Übernahme von Amberpoint zu Oracle und ist heute Bestandteil des WebLogic Management Pack Enterprise Edition und des SOA Management Pack Enterprise Edition. Aus technischer Sicht ist OBTM eine JEE-Anwendung und läuft auf Oracle WebLogic-Servern. Das Deployment kann flexibel an das eigene Lastverhalten angepasst werden. OBTM besteht aus den drei zentralen Servern: btmMain (Hauptprogramm, Web-UI), btmPerformanceServer (Überwachung der Services) und btmTransactionServer (Überwachung der Transaktionen). In einer Oracle Datenbank, der BTM Database, halten die zentralen OBTM-Server ihre Informationen.

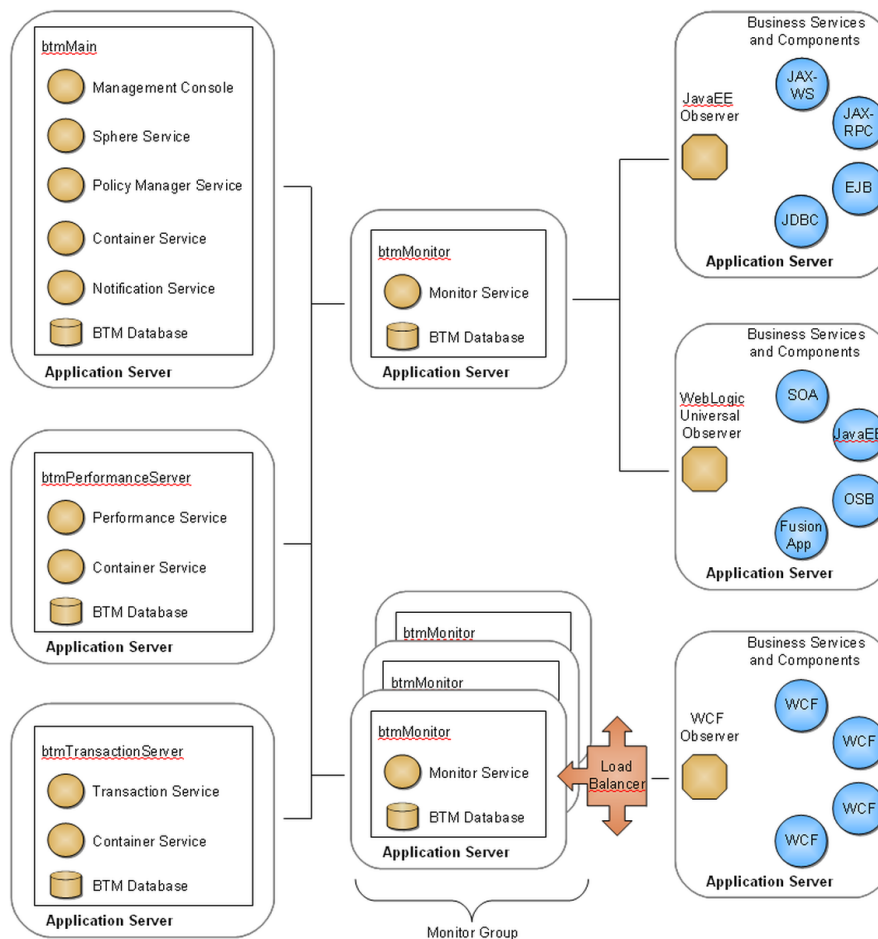


Abb. 1: Architektur des OBTM

Für die Überwachung der Server sind die Observer zuständig. Diese sind Libraries, welche mit den überwachten JEE Containern geladen werden. Sie sammeln die Informationen ein und melden sie an das OBTM. Zwischen den Observern und den zentralen Servern sitzen die Monitor Services. Diese sammeln die Daten von den Observern ein und schreiben sie in die BTM Database. Hier kann auch

mit einer Monitor-Gruppe aus mehreren Server gearbeitet werden, wenn dies aus Skalierungsgründen notwendig ist.

Erste Schritte mit OBTM

Falls man OBTM zunächst ausprobieren möchte, ohne gleich eine komplexe Infrastruktur aufzubauen, kann man zunächst mit der sogenannten "Demo-Installation" starten. In dieser Variante sind die zentralen Server zusammen mit dem Monitor Service in einem Archiv gebündelt, welches am Stück deployed wird. Damit wird die Installation kinderleicht: WLS installieren und Domain aufsetzen, OBTM Demo-Installation deployen, Konfigurationsassistenten durchklicken und fertig. Sogar eine Embedded-Datenbank wird mitinstalliert und fertig konfiguriert.

Diese Variante ist natürlich ausschliesslich für Testzwecke vorgesehen und der Setup-Assistent weist auch ausdrücklich darauf hin. Wenn man aber ohnehin OBTM nur testen möchte, kommt man so in weniger als einer halben Stunde zu einer lauffähigen Umgebung.

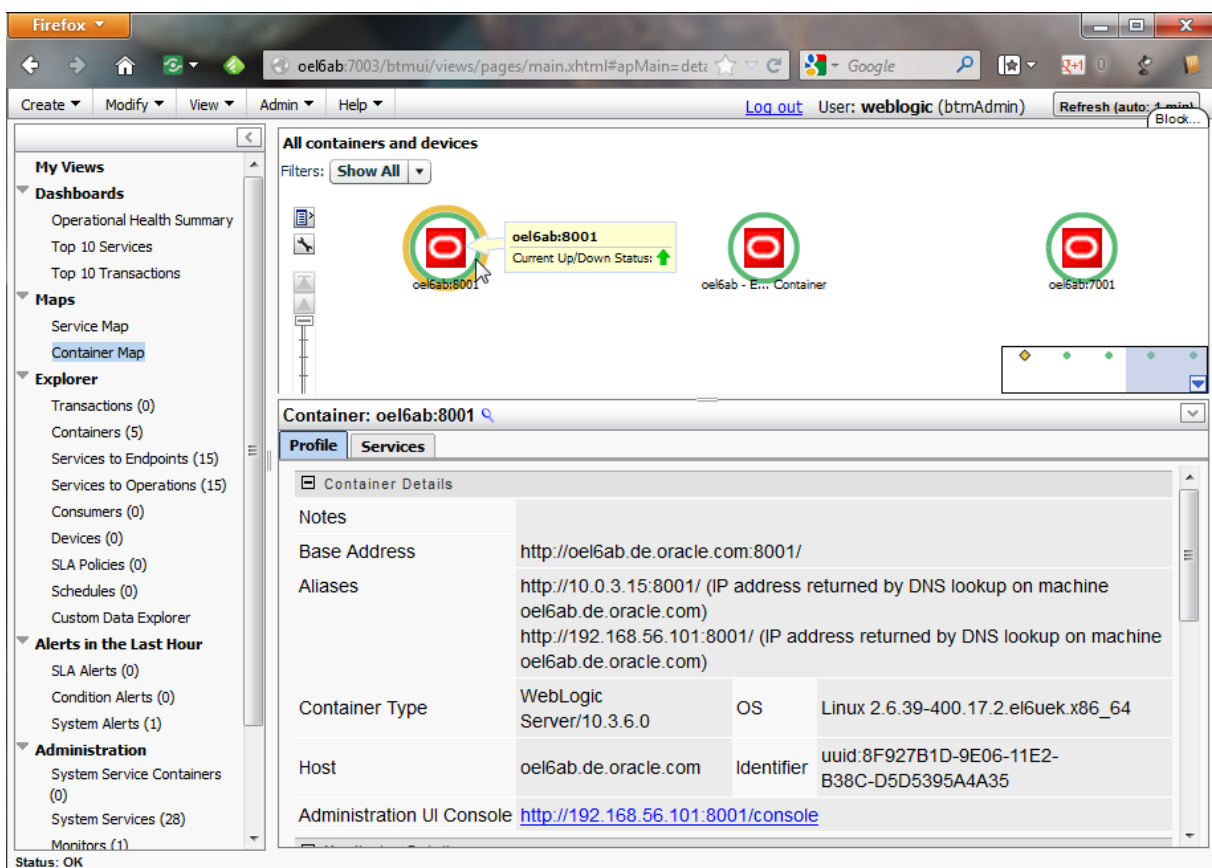


Abb. 2: Erkannte Application Server in der OBTM Web-UI

Nachdem die OBTM-Server laufen, muss Last auf die beobachteten Systeme gebracht werden. OBTM „lernt“ nämlich die Infrastruktur. Durch beobachten der Nachrichten, welche die Systeme untereinander austauschen. Hierzu müssen allerdings noch die Observer installiert werden, welche nicht nur für WebLogic Server zur Verfügung stehen, sondern z.B. auch für JBoss, IBM WebSphere und Microsoft-IIS. Sind die Observer installiert, kann man sich entspannt zurücklehnen und beobachten, wie OBTM die Abhängigkeiten identifiziert. Die beiliegende Web-UI aktualisiert sich standardmässig automatisch, so dass man zusehen kann, wie die Komponenten gefunden werden. Als

erstes wird es die beteiligten Application Server erkennen und ihren Zustand darstellen. Bis hier käme man natürlich auch noch mit klassischem System-Management.

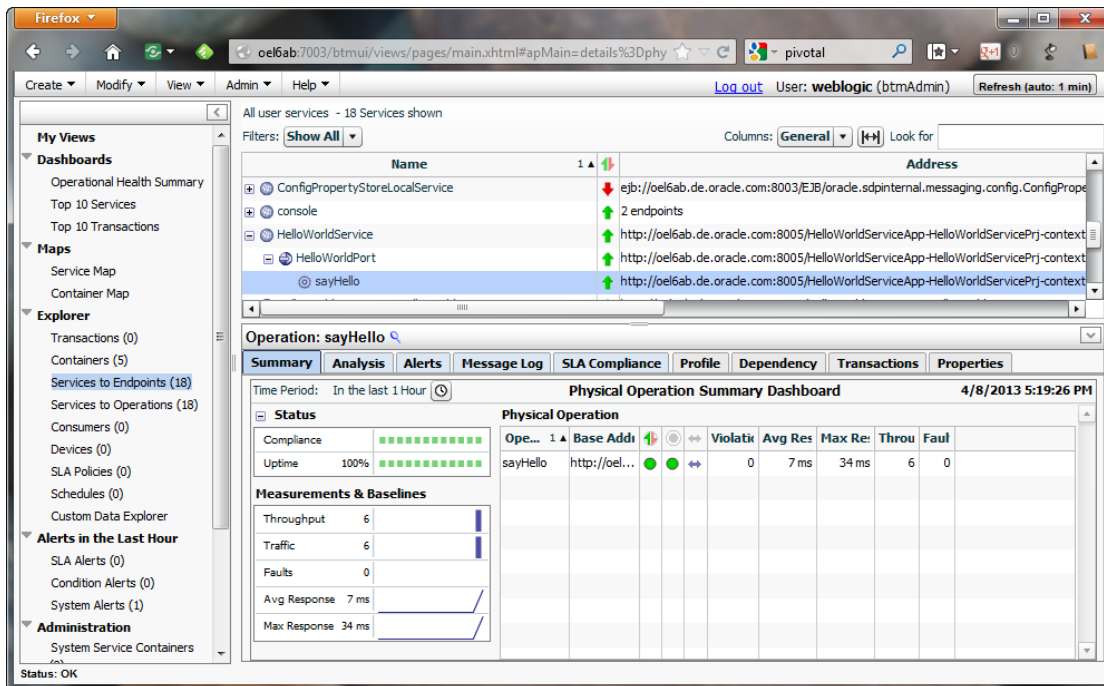


Abb. 3: Erkannte Services in der OBTM Web-UI

Danach wird OBTM die Services entdecken und eine Service-Map aufbauen, welche die Abhängigkeiten unter den Services darstellt. Dies passiert vollautomatisch dadurch, dass OBTM den Nachrichtenverkehr der überwachten Server verfolgt. Nach kurzer Zeit ist bekannt, von welchen Systemen ein Service aufgerufen wird.

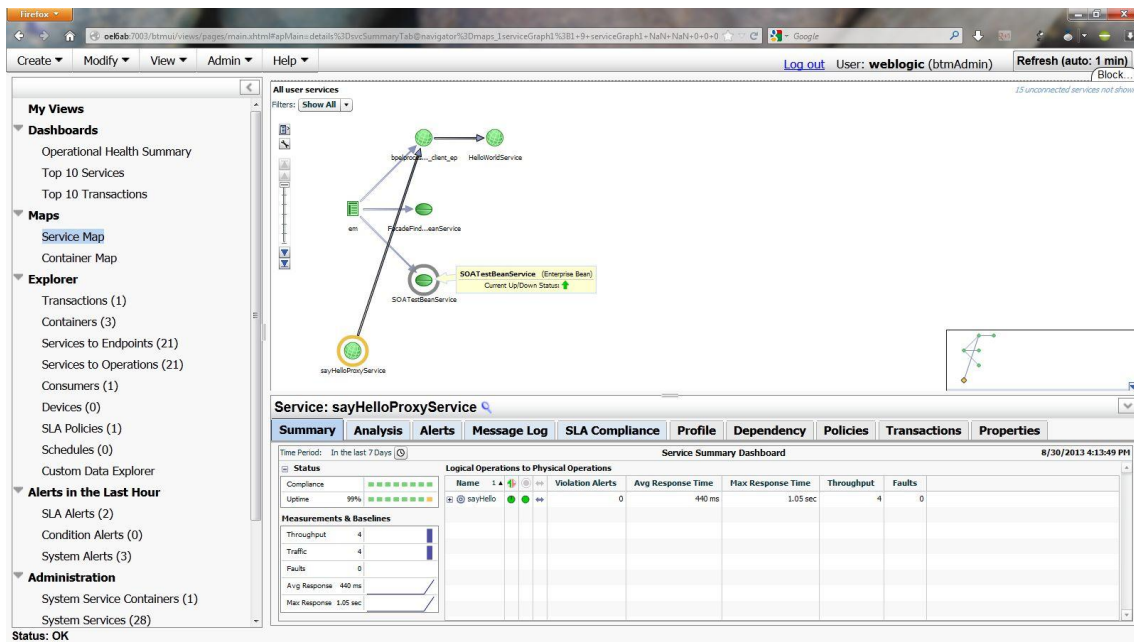


Abb. 4: Erkannte Services in der OBTM Web-UI

Die Service Map ist aber nur eine Basisfunktion von OBTM. Darauf aufbauend lassen sich die namensgebenden Geschäftstransaktionen definieren. Hierzu wird der Einstiegspunkt der Transaktion ausgewählt und die abhängigen Services werden aus der Service Map automatisch übernommen. Sofern nicht alle an der Transaktion beteiligten Services das gleiche Datenmodell nutzen, ist es wahrscheinlich, dass der gleiche Parameter bei jedem Service einen anderen Namen hat. So heisst die Auftragsnummer beispielsweise mal ANR, mal AuftrNr oder noch anders. Will man jetzt herausfinden, warum ein bestimmter Auftrag irgendwo im System hängt, so müsste man eine Suche über alle beteiligten Services starten. OBTM bietet hierfür die Möglichkeit, die Parameter über alle Aufrufe mit standardisierten Namen zu versehen, so dass z.B. global nach ‚Auftragsnummer = 123456‘ gesucht werden kann. Dabei ist hilfreich, dass OBTM auch die Inhalte bereits gelaufener Transaktionen in einem konfigurierbaren Zeitfenster speichert, so dass auch in den bereits durchgelaufenen Transaktionen gesucht werden kann.

Erste Schritte mit OBTM

Im Rahmen des Vortrags wird das OBTM anhand eines Beispiels vorgeführt. Wer danach interesse hat, OBTM einmal selbst auszuprobieren findet das komplette Beispiel als Handson-Tutorial auf meinem BLOG (s.u.).

Kontaktadresse:

Arne Brüning
Oracle Deutschland B.V.
Kühnehöfe 5
D-22761 Hamburg

Telefon: +49 (0) 40-89091 220
Fax: +49 (0) 40-89091 250
E-Mail: arne.bruening@oracle.com
Internet: www.oracle.de
fmw-deutsch.blogspot.com