

Глубокое погружение в RAC для разработчиков впечатления слушателя семинара

14-15 февраля прошел первый выпуск двух дневного семинара **“Глубокое погружение в RAC для разработчиков”**. Я говорю “первый”, поскольку желающих посетить этот семинар набралось в три раза больше, чем могла вместить аудитория, и организаторы семинара обещали его повторить примерно через полтора месяца.

Приглашение на семинар было, пожалуй, одним из самых необычных из всех анонсов и извещений о мероприятиях, которые мне было дано посетить. Обращение организаторов к будущим слушателям содержало не только концепцию и программу семинара, но и квалификационные требования к слушателям, а также возможные задачи и работы после проведения этого тренинга. Судите сами по следующему отрывку (с полным текстом документа можно познакомиться, например, здесь - <http://www.dsvolk.ru/oracle/events/2008-02-14/>).

“Дорогие **технические архитекторы, консультанты, администраторы**, этот семинар для Вас, [поскольку предполагается ответить на вопросы]:

- Заказчик давно просит Вас перенести приложение в RAC, но в команде разработчиков не хватает критической массы знаний этой технологии?
- Ваше приложение написано 18 лет назад и плохо масштабируется. Пора выпускать новую версию, но с технологией RAC у Вас никто не работал?
- О чем надо в первую очередь подумать, чтобы быть уверенным в качественном результате проекта миграции на RAC? Вам нужен Roadmap to RAC?

Присоединяйтесь к разработчикам, чтобы сделать первый шаг к реальному проекту. Мы обещаем, что после семинара Вам захочется отложить текущие дела и начать делать первые шаги ко кластеризации Ваших приложений.”



**“RAC уже
свистнул”**

Вступительное слово руководителя семинара Сергея Данилова практически повторило его статью **“Семинар “Глубокое погружение в RAC для разработчиков”**, которая печатается в этом выпуске журнала. Поэтому в репортаже я соединю выдержки из его вступительных слов в первый и во второй дни семинара. Сергей отметил, что существует конфликт интересов между тем, что удобнее приложению, и тем, что удобнее Заказчику, и семинар как раз задуман именно для того, чтобы разрешить этот конфликт интересов. Он сказал, что, с его точки зрения, объяснить простыми словами Заказчику, что такое RAC, можно следующим образом:

“...RAC – это “технология виртуализации”, объединяющая вычислительную мощность нескольких компьютеров для достижения высокой готовности, масштабируемости и оптимальной стоимости эксплуатации приложений.....

RAC создает для приложений “иллюзию” того, что они работают на одном большом компьютере. В реальности же приложения функционируют на целом наборе компьютеров...

Виртуализация – это отказ от жесткой связи между приложением и машиной, на которой оно работает. Применяя RAC можно запускать приложение в разных вычислительных средах и таким образом уйти от недублируемых точек отказа, получить дополнительную масштабируемость, понизить стоимость всего проекта.”

Сергей задал и ответил на очень важный вопрос **“Чего НЕТ на этом семинаре”**:

- Проектирование аппаратного комплекса для установки clusterware.
- Установка clusterware, установка БД, добавление БД в кластер.
- Деталей работы ASM, серверных процессов, передачи данных по interconnect и пр.
- Настройка производительности (Performance tuning) RAC
- Поиск и выявление узких мест (troubleshooting) RAC

За то на вопрос **“Когда надо использовать RAC”** был дан довольно исчерпывающий ответ:



1. Когда приложения перерастают даже самые большие компьютеры, то есть не хватает масштабируемости.
2. Если большие компьютеры слишком дороги и не хватает средств. (Один процессор может стоить \$30,000-\$50,000)
3. Когда один большой компьютер представляет собой “недублированную точку отказа” (SPOF) и не хватает высокой готовности

Третий очень важный вопрос вопрос и ответ ведущего **“Когда НЕ стоит использовать RAC”**:
Когда приложение:

- Плохо масштабируется даже на одном компьютере
- Использует устаревшее API dbms_pipe, dbms_alert и т.п.
- Плохо спроектировано

Интересная закономерность: **“RAC обладает “эффектом усиления”**:

- Масштабируемое приложение будет работать хорошо
- Плохо масштабируемое приложение будет работать еще хуже
- Плохо спроектированное приложение вообще работать не будет “



Как эти вопросы решаются в практике применения RAC и в разработке программ и были посвящены презентации этого уникального семинара. Надо отметить, что пересказать все или хоть часть презентаций совершенно невозможно. Технические сведения и рекомендации содержались как в докладах и “живых” демонстрациях, так и в ответах на вопросы аудитории и в стихийно возникавших небольших дискуссиях. Несколько самых пригодных для репортажа цитат. Сразу же прошу извинить, что не указываю авторов презентаций, откуда берутся эти цитаты и рисунки. Думаю, что не ошибусь, если назову эту информацию общим достоянием всех лекторов, а теперь и всех слушателей семинара. Итак:

- **Нам необходимо:**

- о координировать работу узлов - **Clusterware**
- о синхронизовать общий кэш данных – **Cache Fusion**
- о уровень работы с дисковой подсистемой - **ASM**
- о подключать пользователей в кластер – для этого существуют VIP-адреса, которыми управляет clusterware

- **RAC 10g: Интегрированное Clusterware**

- о Oracle предоставляет Clusterware на всех платформах бесплатно
- о Начиная с 10g не требуется дополнительно вендорского clusterware
- о Clusterware может использоваться для защиты не только БД

- **Автоматическое управление рабочей нагрузкой**

- о Представляет собой вид абстракции, используемой клиентами для разделения работы на логические рабочие нагрузки
- о Сервисы – ключевое понятие
- о Каждый сервис представляет собой нагрузку с
 - общей функцией
 - общими пороговыми показателями уровня сервиса
 - общим приоритетом
 - с общим дисковым пространством
 - Например: CRM-система, управление персоналом, система получения отчетности

- **Высокая готовность сервисов в RAC**

- о Сервисы **всегда** доступны на одном или более экземпляров.
- о В случае сбоя экземпляра или узла, другой экземпляр автоматически подключается к обеспечению функционирования сервиса.
- о **Предпочтительные экземпляры:**
 - Экземпляры на которых сервис стартует изначально
 - Определяют начальное количество экземпляров для обслуживания сервиса
- о **Доступные экземпляры:**
 - Используются в случае сбоя предпочтительного экземпляра.

- **Типы распределения нагрузки**

- о Балансировка **во время соединения** становится возможной благодаря процессам listener работающим на всех узлах кластера:
 - Балансировка нагрузки на стороне клиента во время соединения;

- Балансировка нагрузки на стороне сервера во время соединения;

- о Балансировка **во время выполнения** становится возможной благодаря использованию пулов соединений:

- Запросы на обслуживание автоматически равномерно распределяются между пулом соединений
- Неявный кэш соединений в JDBC (встроенная возможность)

- **RAC: крах и восстановление экземпляра**

- о Восстановление экземпляра выполняется другими узлами кластера автоматически
- о Global Enqueue Services (GES) перераспределяет enqueues
- о Global Cache Services (GCS) processes remaster only those resources that lose their masters
- о SMON определяет блоки, которые необходимо восстановить, возможно производит их “сборку” из buffer cache других экземпляров.

- **Общие принципы разработки приложения для среды Oracle RAC**

- о Нет никаких специальных принципов и практик по разработке приложений для RAC
- о Если приложение не масштабируется на одном узле – оно не будет быстрее работать в кластере
- о Плохо написанные SQL не станут работать лучше в кластере
- о Стандартные решения по оптимизации SQL и схеме данных решают > 80% проблем с производительностью
- о Убедитесь, что Ваше приложение не использует не поддерживаемых в среде RAC пакетов:
 - dbms_pipe (только в рамках одного узла)
 - dbms_alert
 - dbms_job (не переноситься на другие узлы после failover)
- о Научите Ваше приложение “отрабатывать” потерю одного из узлов кластера
- о Партиционируйте приложение
 - OLTP пользователи на одном узле
 - Отчеты на другом
- о Задачи, требующего высокого конкурентного доступа к общим данным собирайте на одном узле

И так далее, и намного-намного больше. Прошло всего несколько дней после семинара, и уже на широко известном сайте **SQL.RU** развернулся обмен мнениями и впечатлениями. Думаю, что многие-многие посетители этого сайта также будут ждать повторения этого незабываемого семинара.

В одном из перерывов второго дня преподаватели и слушатели семинара сфотографировались на память. (Спасибо компании РДТЕХ за разрешение воспользоваться этим снимком. Спасибо, Инга, за отличное фото.)



В заключение репортажа приведу RAC-омедации
руководителя семинара Сергея Данилова:

- Поскольку еще **нет критической массы специалистов по RAC**,

необходима систематическая передача знаний
ключевым группам разработчиков. Для них это должно
быть бесплатно. Одной из форм такого взаимодействия
является данный семинар «Глубокое погружение в RAC
для разработчиков» (RAC DD4D);

- Мы часто приходим к Заказчику, когда **уже поздно**.
Надо работать с Заказчиком, начиная от Hardware-
сайзинга.

Надо постараться прийти к заказчику до Hardware-
вендора, либо, что еще лучше, вместе с ним.

Надо вести Заказчика через весь жизненный цикл
внедрения RAC и не оставлять Заказчика один на один с
RAC.

- Сейчас еще наблюдается **нехватка историй успеха**.

Надо избавляться от “паранойи по безопасности” и
больше, теснее работать с людьми.

Анатолий Бачин,
“Oracle Magazine/Русское Издание”

**Наши цели ясны,
задачи определены,
RAC свистнул и
кушать уже подано.
За работу,
дорогие oRACle-исты!**

