

Процессы в распределенных системах

Потоки выполнения в нераспределенных системах

СПО ВМ фарева ;-)

Keywords:

process, process table, MMU, TLB, swapping, thread, IPC,
context switching,

Потоки выполнения в распределенных системах

Клиенты/серверы.

Клиенты: однопоточные, многопоточные.

Серверы: однопоточные, многопоточные, машина состояний.

Keywords: sync/async, передача данных.

Перенос кода

Традиционно перенос кода в РС происходит в форме переноса процессов (process migration), где процесс целиком переносится с одной машины на другую.

Это дорогостоящая и сложная задача, которая тем не менее может применяться для увеличения производительности РС.

Причины для миграции

1. Производительность

Может увеличиться при переносе с сильно загруженной машины на слабо загруженную.

Эта задача пока второстепенна, по сравнению со *снижением коммуникационного трафика*

Причины для миграции

2. Уменьшение коммуникационного трафика.

Перенос обрабатывающего кода ближе к месту
нахождения данных.

Например при работе с БД.

Причины для миграции

3. Увеличение производительности за счет параллелизма.

Сравнительно простая программа работающая на множестве машин.

Например SETI@Home.

Причины для миграции

4. Гибкость.

Возможность динамической конфигурации расположения обрабатывающих программ.

Шаблон процесса в распределенной системе

- Сегмент кода
- Сегмент ресурсов
- Сегмент исполнения

Модели переноса кода: по степени мобильности

Модель со слабой мобильностью (weak mobility) — абсолютный минимум для переноса кода.

Модель с сильной мобильностью (strong mobility) — характерная черта: работающий процесс может быть приостановлен, перенесен на другую машину и выполнение будет продолжено с места останова.
Возможность клонирования процесса.

Модели переноса кода: по инициатору

Инициатор — отправитель
(Пример: загрузка своих программ на сервер)

Инициатор — получатель
(Пример: java-апплет)

- * Не зависит от степени мобильности.
- ** Проблемы с защитой.

Варианты переноса кода: слабая мобильность

- **Исполнение в процессе-приемнике**
- **Исполнение в отдельном процессе**

Варианты переноса кода: сильная мобильность

- Перенос процесса
- Клонирование процесса

Перенос и локальные ресурсы

Типы связи процесса с ресурсами

- привязка по идентификатору
 - привязка по значению
 - привязка по типу

Типы связи ресурса и машины

- неприсоединенные ресурсы
 - связанные ресурсы
- фиксированные ресурсы

Варианты переноса кода с учетом привязки ресурсов

	Неприсоединенный ресурс	Связанный ресурс	Фиксированный ресурс
По идентификатору	MV (GR)	GR (MV)	GR
По значению	CP (MV, GR)	GR (CP)	GR
По типу	RB (MV,CP)	RB (GR, CP)	RB (GR)

GR — организовать глобальную ссылку

MV — перенести ресурс

CP — скопировать значение ресурса

RB — выполнить новую привязку процесса к ресурсу

Программные агенты

Программный агент (software agent) – автономный процесс, *способный реагировать на среду исполнения и вызывать изменения в среде исполнения*, возможно в кооперации с пользователем или другими агентами.

Однозначного определения не существует.

Классификация агентов

Единой классификации не существует.

Выделяют:

- кооперативные агенты
 - мобильные агенты
 - интерфейсные агенты
- информационные агенты

Важнейшие свойства агентов

Свойство	Общность для агентов	Описание
автономность	да	способность функционировать самостоятельно
реактивность	да	Способность своевременно реагировать на изменения в своем окружении
проактивность	да	Способность инициировать действия, влияющие на окружение
коммуникативность	да	Способность обмениваться информацией с пользователями и другими агентами
продолжительность	нет	Относительно долгое время жизни
мобильность	нет	Способность перемещаться с места на место
адаптивность	нет	Способность к обучению

Обобщенная модель платформы агента согласно FIPA

