ОПТИМИЗАЦИЯ ПОДСИСТЕМЫ ПАМЯТИ ЯДРА ОС LINUX ДЛЯ ВК ЭЛЬБРУС-3S С ПОДДЕРЖКОЙ NUMA

Кравцунов Е.М.
Научный руководитель: д.т.н., проф. Семенихин С.В.
ОАО «ИНЭУМ им. И.С. Брука»
119334, Москва, ул. Вавилова, д. 24
Тел.: (499) 135-21-48; e-mail: kravtsunov_e@mcst.ru

NUMA (Non Uniform Memory Access) – это способ организации взаимодействия процессора с памятью, при котором время доступа к разным областям основной памяти различается. Процессоры в NUMA платформе группируются в узлы, содержащие один или несколько процессоров. Каждому узлу ставится в соответствие область основной памяти, время доступа к которой для процессоров узла является минимальным. Время доступа процессора к памяти соседнего узла существенно больше времени доступа к памяти своего узла. Процессор имеет доступ к областям памяти всех узлов. Когерентность кэш-памяти процессоров в NUMA платформах поддерживается аппаратно.

Многопроцессорный вычислительный комплекс Эльбрус-3S является NUMA платформой. Эльбрус-3S включает в себя 4 узла NUMA. Каждый узел состоит из одного процессора Эльбрус-3S. Каждый узел связан с соответствующей областью памяти по отдельной шине. Взаимодействие процессора с памятью другого узла осуществляется с использованием процессорных линков, которыми связаны между собой узлы в Эльбрус-3S.

Средняя производительность вычислительного комплекса существенно зависит от природы исполняемых задач, которые в этом смысле можно разделить на 3 типа:

1. задачи, чувствительные к задержкам, которые связаны с обращениями в память; как правило, плохо распараллеливаются по памяти. Для повышения производительности в этом случае необходимо выделять память в том же узле, где выполняется задача;

2. задачи чувствительные к полосе пропускания; такие задачи хорошо распараллеливаются по памяти. Примером является параллельная запись в память большого количества блоков данных. Для таких задач решающим с точки зрения производительности является полоса пропускания – количество процессоров, совместно выполняющих эту работу. Если процессоров много, то задержка доступа в память, связанная с выделением страниц на чужом узле не оказывает существенного влияния на среднюю производительность;

3. задачи смешанного типа.

Ядро ОС Linux представляет собой интерактивную задачу, которую можно отнести к первому типу. Поэтому для повышения средней производительности ОС на ВК Эльбрус-3S эффективным является использование алгоритма создания копий исполняемого кода и константных данных ядра на каждом узле NUMA ВК Эльбрус-3S.

Оптимизация возможна, благодаря следующей особенности реализации ядра ОС Linux применительно к архитектуре «Эльбрус»: образ ядра загружается в собственное виртуальное пространство, то есть для кода, модулей и данных ядра, доступных только по чтению, выделен специальный диапазон виртуальных адресов. В ОС Linux все виртуальное пространство разделено на две части: виртуальные адреса, меньшие значения TASK_SIZE

могут использоваться в режиме пользователя, виртуальные адреса со значениями выше TASK_SIZE предназначены для ядра. В варианте архитектуры «Эльбрус» виртуальное пространство, предназначенное для ядра, также разделено на 5 непересекающихся диапазонов адресов: 1) диапазон для мапирования физической памяти, 2) диапазон для мапирования образа ядра и модулей, 3) диапазон для мапирования аппаратных и пользовательских стеков, 4) диапазон для мапирования пространств ввода-вывода. 5) диапазон для мапирования таблиц страниц. Диапазон для мапирования образа ядра и модулей начинается со значения адреса VM_KERNEL_BASE и имеет размер kernel_image_size = text" + "nodedata" + "data" + "bss", где размеры областей text, nodedata, data, bss – вычисляются из значений соответствующих меток в объектном файле ядра vmlinux.

Создание копий производится для страниц исполняемого кода и данных ядра, доступных только по чтению, адрес которых принадлежит диапазону виртуальных адресов от VM_KERNEL_BASE до KERNEL_NODEDATA_END, где KERNEL_NODEDATA_END – это последний адрес сегмента nodedata (или начальный адрес сегмента data). Копирование производится при начальной загрузке ядра после получения ядром управления от программы начальной загрузки (bootloader) и до переключения на виртуальную адресацию. Копирование производится с узла, которому принадлежит ведущий процессор (bootstrap) на остальные узлы. Копирование производится по физическим адресам. После копирования и до переключения на работу по виртуальным адресам на всех узлах создаются копии таблиц страниц. Эти таблицы после переключения на виртуальную адресацию используются для трансляции адресов. После построения таблиц для любого адреса из диапазона [VM_KERNEL_BASE; KERNEL_NODEDATA_END] на каждом узле NUMA системы однозначно определен физический адрес. Ясно, что после создания копий физические адреса, соответствующие одному и тому же виртуальному адресу из указанного диапазона, различаются. После успешного завершения построения таблиц страниц на всех узлах происходит переключение на виртуальную адресацию, после чего на bootstrap процессоре вызывается архитектурно-независимая функция start_kernel().

Реализация этого алгоритма позволила уменьшить среднее время активизации процесса на 40% для ВК Эльбрус-3S с поддержкой NUMA. Такое существенное уменьшение времени активизации процесса положительно сказывается на средней производительности ВК в целом.