В этой версии задания используется библиотека MPJ Express, являющаяся Java-оберткой над MPI. При старте приложения загружается конфигурация, в которой указано, на сколько строк и столбцов разбивается пространство моделирования. Допустим, это x столбцов и y строк. Тогда на каждый столбец выделяется y + 1 процессов, которые объединяются общим коммуникатором. Процесс, имеющий в полученном коммуникаторе ранг 0, становится "метаклеткой", остальные же - обычными клетками. Процесс обсчета происходит в обычных клетках, при этом каждая обычная клетка знает о агентах из верхней и нижней клеток, находящихся на расстоянии не более delta. Общение с клетками слева и справа происходит посредством метаклеток. Метаклетки объединяются своим коммуникатором.

(синие прямоугольники – простые клетки, красные – метаклетки)

Благодаря такой организации пространства появляется возможность перемещать границы клеток в каждом столбце независимо от другого, и тем самым балансировать нагрузку в пределах столбца. Изменяя ширину столбца, можно перераспределять нагрузку между столбцами.

(пример перераспределения)

Обычная клетка состоит из пяти частей: собственно клетка, верхний, нижний, левый и правый края. Какие-либо действия производятся только над агентами из основной части, края используются только для поиска пары для размножения. Ширина каждого края - delta, то есть клетка "знает" о своих соседях на расстоянии delta с каждой из сторон.

(светло-серые прямоугольники – края, темно-серый – основная часть клетки

Между итерациями происходит синхронизация. Есть два вида синхронизации: вертикальная и горизонтальная. При вертикальной синхронизации каждая клетка отправляет в клетку, находящиеся выше ее в столбце, агентов, находящихся выше, чем topY+delta (точка отсчета координат находится в левом верхнем углу, ось ординат направлена вниз) а в нижнюю - тех, что ниже, чем bottomY-delta (то есть тех, кто вышел за пределы клетки в соответствующем направлении, и ту часть основной части, которая будет соответствующим краем соседа).

Затем она, соответственно, получает агентов, переданных ей соседями. Передача происходит асинхронно (с помощью MPI\_Isend/MPI\_Irecv), поэтому циклов ожидания не вознкает. Так как клетка может передать только агентов, известных ей на данной итерации, то из этого возникает необходимость того, чтобы высота клетки была не меньше 2\*delta (в противном случае есть вероятность, что клетка снизу не получит информацию об агенте, пришедший сверху в эту же синхронизацию).

Горизонтальная синхронизация происходит между метаклетками. Сначала каждая метаклетка получает от каждой клетки агентов, вышедших за ее левый и правый края или приблизившихся к ним на расстояние не больше delta (вышедших за пределы и тех, кто виден из соседнего столбца). Затем она пересылает соответствующих агентов метаклеткам левее и правее ее в сетке. Далее каждой обычной клетке из столбца метаклетки отсылаются видимые ей агенты (происходит разбиение агентов на уровни по высотам).

Выбор агентов для пересылки в каждом из направлений происходит на основе размеров клетки. Исходя из этого, балансировка может быть реализована простым изменением размера клеток. После изменения размеров клетка отошлет уже не принадлежащих ей агентов соседям, а те их примут.

Расчет размеров может производиться по вариации алгоритма Брезенхема: каждая клетка отсылает метаклетке распределение агентов по высотам, та строит общее распределение по столбцу и находит среднее количество агентов в клетке. Затем для каждой клетки рассчитываются размеры так, чтобы количество агентов в ней после синхронизации было максимально приближено к среднему. К сожалению, вышеописанный алгоритм синхронизации накладывает условие, что новая нижняя граница не может быть выше старой верхней + delta (чтобы не пришлось пересылать агентов через клетку), а новая верхняя, в свою очередь, ниже старой нижней - delta.

Вывод данных реализуется с помощью графической библиотеки Swing, и под него используется отдельный процесс. Каждая из клеток отправляет этому процессу своих агентов, а тот, в свою очередь, их получает и обрабатывает, проставляя в таблицу – модель дисплея – фенотипы агентов. В графическом потоке работает таймер, который через заданные промежутки времени считывает эту таблицу и обновляет изображение, исходя из считанных данных.