

Example of a Malleability application pattern

Alberto Cascajo, Javier Fernández, David E. Singh

October 14-16, 2026, Madrid, Spain

38th IEEE/SBC International Symposium on Computer Architecture and High-Performance Computing

ARCOS
COMPUTER ARCHITECTURE, COMMUNICATIONS AND SYSTEMS RESEARCH GROUP

Outline

- Application pattern example (C-based)
  - original pattern
  - malleable pattern
- Important modified sections
  - Check process type
  - Re/synchronize initial data
    - Initial section
    - Loop section
  - Synchronize iterations
  - Set user hints
  - Check process status

Original application pattern: C-based

```
int main(int argc, char *argv[]){
  …
  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &mpi_rank);
  MPI_Comm_size(MPI_COMM_WORLD, &mpi_size);

  <syncronize initial data: native proceses>

  while (it <itmax) {

    <Compute and distribute data>

    it++;
  }
  MPI_Finalize();
}
```

Malleable application pattern: C-based

```
int main(int argc, char *argv[]){
  …
  MPI_Init(&argc, &argv);
  MPI_Comm_rank(ADM_COMM_WORLD, &mpi_rank);
  MPI_Comm_size(ADM_COMM_WORLD, &mpi_size);

  <Check process type: native or spawned>
  <syncronize initial data: native and spawned proceses>
  <Syncronize iterations among processes>

  ADM_MonitoringService (ADM_SERVICE_START);

  while (it < itmax) {
      …
      …
```

Malleable application pattern: C-based

```
…
while (it < itmax) {
  <Set user malleability hints>
  <Resynchronize initial data with spawned processes>

  ADM_MalleableRegion (ADM_SERVICE_START);

  <Compute and distribute data>

  it++:

  status = ADM_MalleableRegion (ADM_SERVICE_STOP);

  <Check process status: removed or active>

}
  ADM_MonitoringService (ADM_SERVICE_STOP);
  MPI_Finalize();
}
```

Modified sections: Check process type

- Uses Malleable attribute ADM_GLOBAL_PROCESS_TYPE

```
…
ADM_GetSysAttributesInt(”ADM_GLOBAL_PROCESS_TYPE", &type);
if (type == ADM_NATIVE) {
  <Process is native>
} else if (type == ADM_SPAWNED) {
  <Process is spawned>
}
…
```

Modified sections: Re/synchronize initial data

- Initial section (mixed with check process type)

```
…
ADM_GetSysAttributesInt(”ADM_GLOBAL_PROCESS_TYPE", &type);
if (type == ADM_NATIVE) {

  <init and redistribute initial data>

} else if (type == ADM_SPAWNED) {

  <Get actual versión of the initial data>

}
…
```

Modified sections: Re/synchronize initial data

- Loop section

```
…
int last_mpi_size = mpi_size
while (it < itmax) {
  …
  if (last_mpi_size != mpi_size) {
    <Send actual versión of the initial data>
  }
  …
```

Modified sections: Synchronize iterations

- Iteration must be updated just before stopping malleable region.

```
…
ADM_RegisterSysAttributesInt(”ADM_GLOBAL_MAX_ITERATION",
                                &itmax);
ADM_GetSysAttributesInt(”ADM_GLOBAL_ITERATION", &it);
…
while (it < itmax) {
  …
  ADM_RegisterSysAttributesInt(”ADM_GLOBAL_ITERATION", &it);
  status = ADM_MalleableRegion (ADM_SERVICE_STOP);
  …
```

Modified sections: Set user hints

- Hints are reset after starting every malleable region

```
…
while (it < itmax) {
  …
  if (<condition to spawn/remove>) {
    ADM_RegisterSysAttributesInt(
                  ”ADM_GLOBAL_HINT_NUM_PROCESS", &proc_hint);
    ADM_RegisterSysAttributesInt(
                  ” ADM_GLOBAL_HINT_EXCL_NODES", &excl_hint);
  }
  …
```

Modified sections: Check process status

- Iteration must be updated just before stopping malleable region.

```
…
while (it < itmax) {
  …
  status = ADM_MalleableRegion (ADM_SERVICE_STOP);
  if (status == ADM_ACTIVE) {
    MPI_Comm_rank(ADM_COMM_WORLD, &mpi_rank);
    MPI_Comm_size(ADM_COMM_WORLD, &mpi_size);
  } else if (status == ADM_REMOVED) {
    break;
  }
  …
```



Example of a Malleability application pattern

Alberto Cascajo, Javier Fernández, David E. Singh

October 14-16, 2026, Madrid, Spain

38th IEEE/SBC International Symposium on Computer Architecture and High-Performance Computing

ARCOS
COMPUTER ARCHITECTURE, COMMUNICATIONS AND SYSTEMS RESEARCH GROUP