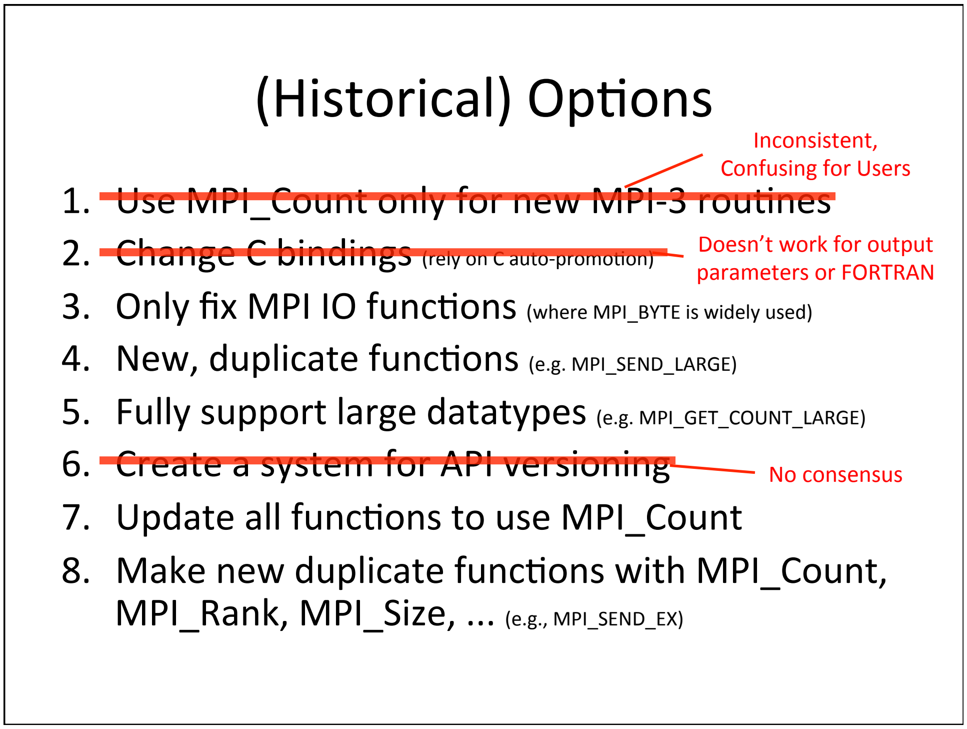
MPI_Count

The Saga Continues



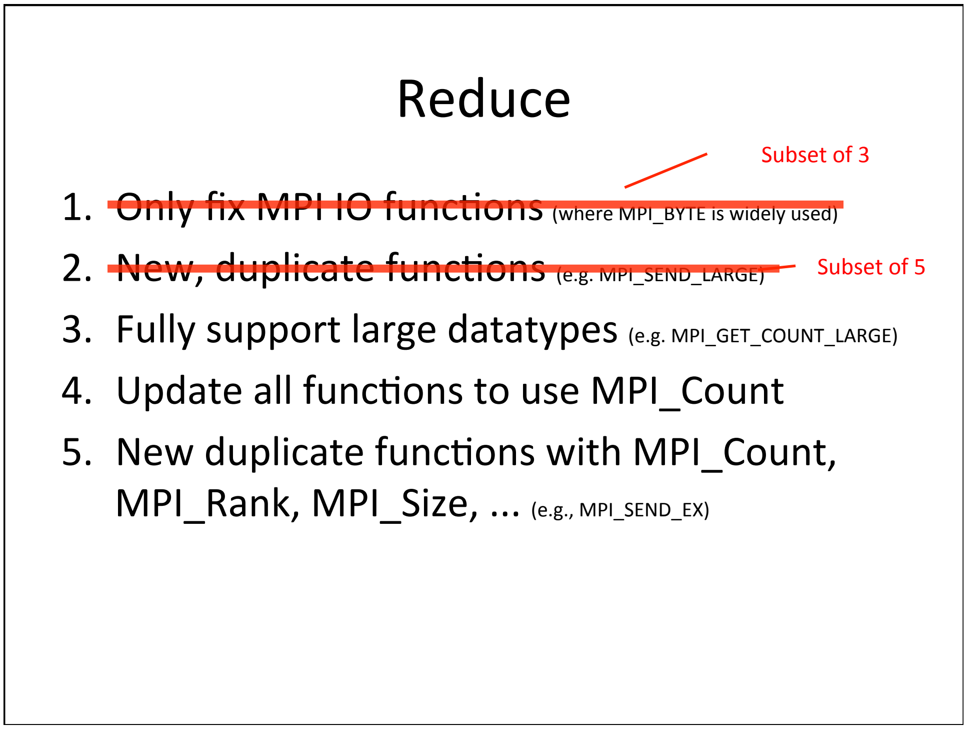
(Historical) Options

1. ~~Use MPI_Count only for new MPI-3 routines~~
2. ~~Change C bindings (rely on C auto-promotion)~~
3. Only fix MPI IO functions (where MPI_BYTE is widely used)
4. New, duplicate functions (e.g. MPI_SEND_LARGE)
5. Fully support large datatypes (e.g. MPI_GET_COUNT_LARGE)
6. ~~Create a system for API versioning~~
7. Update all functions to use MPI_Count
8. Make new duplicate functions with MPI_Count, MPI_Rank, MPI_Size, ... (e.g., MPI_SEND_EX)

Inconsistent, Confusing for Users

Doesn’t work for output parameters or FORTRAN

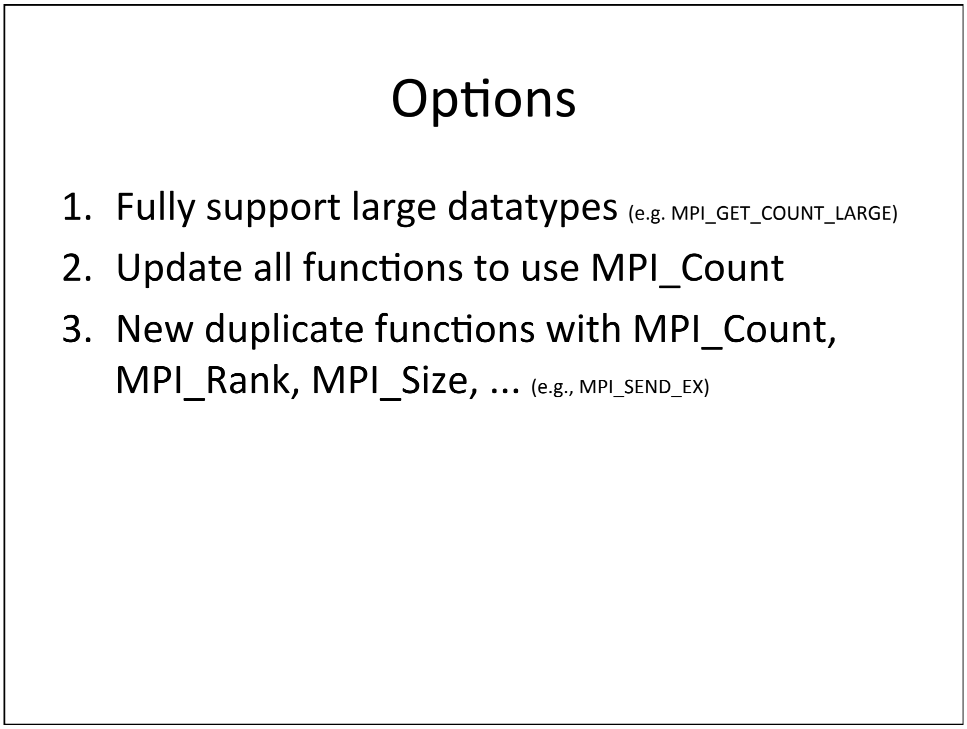
No consensus



Reduce

Subset of 3

1. ~~Only fix MPI IO functions (where MPI_BYTE is widely used)~~
2. ~~New, duplicate functions (e.g. MPI_SEND_LARGE)~~ Subset of 5
3. Fully support large datatypes (e.g. MPI_GET_COUNT_LARGE)
4. Update all functions to use MPI_Count
5. New duplicate functions with MPI_Count, MPI_Rank, MPI_Size, … (e.g., MPI_SEND_EX)



Options

1. Fully support large datatypes (e.g. MPI_GET_COUNT_LARGE)
2. Update all functions to use MPI_Count
3. New duplicate functions with MPI_Count, MPI_Rank, MPI_Size, … (e.g., MPI_SEND_EX)

Fully Support Large Datatypes

- Not as simple as “fix 3 functions and voila!”
- Encodes counts into datatypes
- Need ‘W’ vesions of all collectives:
  - MPI_GATHERW
  - MPI_SCATTERW
  - MPI_ALLGATHERW
  - MPI_IGATHERW
  - MPI_ISCATTERW
  - MPI_IALLGATHERW

Need a mechanism to specify different counts: different datatypes

Use MPI_Count everywhere

- Ticket #224
- Can break backward compatibility
  - Makes all existing MPI codes non-compliant
- Hard to mix int & MPI_Count in the same app

The Whole Enchilada

- New duplicate functions that add everything
  - – MPI_Count
  - – MPI_Size
  - – MPI_Rank
  - – Timeout?
  - – …
- I'd love to see a proposal for this…

Conclusion

- Fixing datatypes to support large counts would happen as part of the 'Whole Enchilada'
  - – Need updated Get/Set_counts, etc
  - – Should have W functions for completeness
- Fixing datatype smallest in scope