

Current Status

- DØ currently uses mc_runjob, written by Greg Graham, Dave Evans and Peter Love for:
 - Official MC production (remote farms)
 - MC production of small samples by users
 - Data reprocessing on remote farms
- mc_runjob is integrated with JIM for official MC production

Expectations

- mc_runjob has turned out to be very useful, but historically grown and needs rewrite + is being adopted by other experiments
- Want the new runjob to be fully integrated into DØ's future production and analysis model:
 - Moving towards using (nearly) the same set of tools for production and user analysis job submission
 - Ultimately want flexibility to run user analysis jobs remotely

- Suite of tools has
 - SAMGrid/JIM (data handling, job submission and monitoring, resource brokerage, etc.)
 - Runjob (workflow, provide monitoring elements)
 - dOrte, sandbox, OS compatibility (run-time environment)
 - d0tools (this is the user interface, for “easy” jobs this will generate the runjob macro)
- Dugan O'Neil is in charge of making sure all of these interact together well, and that the chosen model fulfills our needs (for both production and user analysis)

Input

- Dugan and Greg already discussing architecture in detail. Agreement of principle exists (I believe).
- DØ-specific aspects are being developed by Peter Love (funded by GRIDPP), who will also contribute (to a certain extent) to the general runjob infrastructure

Concerns

- Primary concern is to maintain adequate computing resources for DØ as long as necessary (computing cannot be the reason experiment stops doing analysis)
 - Implies use of resources common with LHC, so need to be compatible
- Runjob-specific constraint related to the fact that it's common between experiments:
 - Careful about what goes in the general pot and what is experiment-specific