NAF feed-back from ILC

Mikael Berggren[1]

[1]DESY
Hamburg

NAF meeting, Helmholtz annual meeting, Dec 2011

Outline

ILC usage

- Test-beam analysis by the Calice R&D collaboration.
- Event generation (Whizard) for MC mass-production of SM - $\mathcal{O}$(10Mevents).
- Full simulation/reconstruction chain for signal samples $\mathcal{O}$(10kevents)
- User analysis of all kinds and sizes.
- Code development.
- NUC members: Steve Aplin, Shaojun Lu.

Coming ILC usage: The DBD

As a companion to the ILC TDR, a "Detailed Baseline Detector document" will delivered by the end of 2012.

- Bench-mark processes at E_{CMS}=1TeV:
  - $e^+e^- \to \nu\bar{\nu}h^0$
  - $e^+e^- \to t\bar{t}h^0$
  - $e^+e^- \to W^+W^-$
- Full simulation, ameliorated wrt. LOI. Assume 1 ab^{-1}.
- Machine backgrounds and same-bunch crossing $\gamma\gamma$ events should be overlaid (in some way...)
- "Relevant" physics backgrounds should be generated.
- Decided to do generation of $e^+e^- \to W^+W^-$ and background on the NAF. (The other two at SLAC and KEK).

Coming ILC usage: The DBD

As a companion to the ILC TDR, a "Detailed Baseline Detector document" will delivered by the end of 2012.

- Bench-mark processes at E_{CMS}=1TeV:
  - $e^+e^- \to \nu\bar{\nu}h^0$
  - $e^+e^- \to t\bar{t}h^0$
  - $e^+e^- \to W^+W^-$
- Full simulation, ameliorated wrt. LOI. Assume 1 ab^{-1}.
- Machine backgrounds and same-bunch crossing $\gamma\gamma$ events should be overlaid (in some way...)
- "Relevant" physics backgrounds should be generated.
- Decided to do generation of $e^+e^- \to W^+W^-$ and background on the NAF. (The other two at SLAC and KEK).

Coming ILC usage: The DBD

As a companion to the ILC TDR, a "Detailed Baseline Detector document" will delivered by the end of 2012.

- Bench-mark processes at E_{CMS}=1TeV:
  - $e^+e^- \to \nu\bar{\nu}h^0$
  - $e^+e^- \to t\bar{t}h^0$
  - $e^+e^- \to W^+W^-$
- Full simulation, ameliorated wrt. LOI. Assume 1 ab^{-1}.
- Machine backgrounds and same-bunch crossing $\gamma\gamma$ events should be overlaid (in some way...)
- "Relevant" physics backgrounds should be generated.
- Decided to do generation of $e^+e^- \to W^+W^-$ and background on the NAF. (The other two at SLAC and KEK).

Coming ILC usage: The DBD

As a companion to the ILC TDR, a "Detailed Baseline Detector document" will delivered by the end of 2012.

- Bench-mark processes at E_{CMS}=1TeV:
  - $e^+e^- \to \nu\bar{\nu}h^0$
  - $e^+e^- \to t\bar{t}h^0$
  - $e^+e^- \to W^+W^-$
- Full simulation, ameliorated wrt. LOI. Assume 1 ab^{-1}.
- Machine backgrounds and same-bunch crossing $\gamma\gamma$ events should be overlaid (in some way...)
- "Relevant" physics backgrounds should be generated.
- Decided to do generation of $e^+e^- \to W^+W^-$ and background on the NAF. (The other two at SLAC and KEK).

Coming ILC usage: The DBD

As a companion to the ILC TDR, a "Detailed Baseline Detector document" will delivered by the end of 2012.

- Bench-mark processes at E_{CMS}=1TeV:
  - $e^+e^- \to \nu\bar{\nu}h^0$
  - $e^+e^- \to t\bar{t}h^0$
  - $e^+e^- \to W^+W^-$
- Full simulation, ameliorated wrt. LOI. Assume 1 ab^{-1}.
- Machine backgrounds and same-bunch crossing $\gamma\gamma$ events should be overlaid (in some way...)
- "Relevant" physics backgrounds should be generated.
- Decided to do generation of $e^+e^- \to W^+W^-$ and background on the NAF. (The other two at SLAC and KEK).

Use case: *WW* generation

- All steps setup to run on the NAF.
  - Integration of all 4 fermion final-states: over-night job, with per mil uncertainty on cross-section
  - Generation of 1 ab^{-1} also over-night job for non-electron final states.
  - STDHEP:s on grid, log-files, steerings, diagram-plots, etc. on the web.
  - Need some automatic error detection.
- Organisation:
  - Hierarchy: ZZ or WW or ZZWWmix / hadronic or leptonic or semi-leptonic / four beam polarisations
  - Separate single boson ($XXee$,$XX\nu_e\nu_e$ or $XXe\nu_e$) final states (t-channel!) from rest.
  - Total number of cases = 36.
- NB: Cross-sections are in the 10 pb^{-1} range $\rightarrow$ we will generate tens of millions of events !!!

Use case: *WW* generation

- All steps setup to run on the NAF.
  - Integration of all 4 fermion final-states: over-night job, with per mil uncertainty on cross-section
  - Generation of 1 ab^{-1} also over-night job for non-electron final states.
  - STDHEP:s on grid, log-files, steerings, diagram-plots, etc. on the web.
  - Need some automatic error detection.
- Organisation:
  - Hierarchy: ZZ or WW or ZZWWmix / hadronic or leptonic or semi-leptonic / four beam polarisations
  - Separate single boson ($XXee$,$XX\nu_e\nu_e$ or $XXe\nu_e$) final states (t-channel!) from rest.
  - Total number of cases = 36.
- NB: Cross-sections are in the 10 pb^{-1} range $\rightarrow$ we will generate tens of millions of events !!!

Use case: *WW* generation

- All steps setup to run on the NAF.
  - Integration of all 4 fermion final-states: over-night job, with per mil uncertainty on cross-section
  - Generation of 1 ab^{-1} also over-night job for non-electron final states.
  - STDHEP:s on grid, log-files, steerings, diagram-plots, etc. on the web.
  - Need some automatic error detection.
- Organisation:
  - Hierarchy: ZZ or WW or ZZWWmix / hadronic or leptonic or semi-leptonic / four beam polarisations
  - Separate single boson ($XXee$,$XX\nu_e\nu_e$ or $XXe\nu_e$) final states (t-channel!) from rest.
  - Total number of cases = 36.
- NB: Cross-sections are in the 10 pb^{-1} range $\rightarrow$ we will generate tens of millions of events !!!

Use case: *WW* generation

- All steps setup to run on the NAF.
  - Integration of all 4 fermion final-states: over-night job, with per mil uncertainty on cross-section
  - Generation of 1 ab^{-1} also over-night job for non-electron final states.
  - STDHEP:s on grid, log-files, steerings, diagram-plots, etc. on the web.
  - Need some automatic error detection.
- Organisation:
  - Hierarchy: ZZ or WW or ZZWWmix / hadronic or leptonic or semi-leptonic / four beam polarisations
  - Separate single boson ($XXee$,$XX\nu_e\nu_e$ or $XXe\nu_e$) final states (t-channel!) from rest.
  - Total number of cases = 36.
- NB: Cross-sections are in the 10 pb^{-1} range $\rightarrow$ we will generate tens of millions of events !!!

Observations about the NAF

- Extremely useful to be able to work interactively in an environment (almost) identical to the batch system:
  - Develop and debug.
  - Write and test scripts.
  - Add a few #$:s to the script.
  - qsub!
- Would not have been possible to so quickly set up the massive generation without this infra-structure.
- Large disk space from LUSTRE is very useful. Compare
  - User AFS quota.
  - User space on Grid node.
- That Lustre is shared gives the users the possibility to do punctual multi-TB work - Test-beam analysis !
- Up to now, this works without imposing quotas. Hope it can stay that way !

Observations about the NAF

- Extremely useful to be able to work interactively in an environment (almost) identical to the batch system:
  - Develop and debug.
  - Write and test scripts.
  - Add a few #$:s to the script.
  - qsub!
- Would not have been possible to so quickly set up the massive generation without this infra-structure.
- Large disk space from LUSTRE is very useful. Compare
  - User AFS quota.
  - User space on Grid node.
- That Lustre is shared gives the users the possibility to do punctual multi-TB work - Test-beam analysis !
- Up to now, this works without imposing quotas. Hope it can stay that way !

Observations about the NAF

- Extremely useful to be able to work interactively in an environment (almost) identical to the batch system:
  - Develop and debug.
  - Write and test scripts.
  - Add a few #$:s to the script.
  - qsub!
- Would not have been possible to so quickly set up the massive generation without this infra-structure.
- Large disk space from LUSTRE is very useful. Compare
  - User AFS quota.
  - User space on Grid node.
- That Lustre is shared gives the users the possibility to do punctual multi-TB work - Test-beam analysis !
- Up to now, this works without imposing quotas. Hope it can stay that way !

Observations about the NAF

- Extremely useful to be able to work interactively in an environment (almost) identical to the batch system:
  - Develop and debug.
  - Write and test scripts.
  - Add a few #$:s to the script.
  - qsub!
- Would not have been possible to so quickly set up the massive generation without this infra-structure.
- Large disk space from LUSTRE is very useful. Compare
  - User AFS quota.
  - User space on Grid node.
- That Lustre is shared gives the users the possibility to do punctual multi-TB work - Test-beam analysis !
- Up to now, this works without imposing quotas. Hope it can stay that way !

Observations about the NAF

- Extremely useful to be able to work interactively in an environment (almost) identical to the batch system:
  - Develop and debug.
  - Write and test scripts.
  - Add a few #$:s to the script.
  - qsub!
- Would not have been possible to so quickly set up the massive generation without this infra-structure.
- Large disk space from LUSTRE is very useful. Compare
  - User AFS quota.
  - User space on Grid node.
- That Lustre is shared gives the users the possibility to do punctual multi-TB work - Test-beam analysis !
- Up to now, this works without imposing quotas. Hope it can stay that way !

Observations about the NAF

- Convenient to access AFS and GRID storage elements.
  - Both Calice and ILC have pre-installed software on AFS $\Rightarrow$
  - Same SW on your desk-top and on the NAF.
  - Corollary: It is preferable that the NAF avoids being on the bleeding edge: Don't be the first to go to the latest-and-greatest SL-version, but upgrade when a sizable fraction of the community has !
- Useful to have access to GRID SE (DESY dcache).

Observations about the NAF

- Convenient to access AFS and GRID storage elements.
  - Both Calice and ILC have pre-installed software on AFS $\Rightarrow$
  - Same SW on your desk-top and on the NAF.
  - Corollary: It is preferable that the NAF avoids being on the bleeding edge: Don't be the first to go to the latest-and-greatest SL-version, but upgrade when a sizable fraction of the community has !
- Useful to have access to GRID SE (DESY dcache).

Observations about the NAF

- Convenient to access AFS and GRID storage elements.
  - Both Calice and ILC have pre-installed software on AFS $\Rightarrow$
  - Same SW on your desk-top and on the NAF.
  - Corollary: It is preferable that the NAF avoids being on the bleeding edge: Don't be the first to go to the latest-and-greatest SL-version, but upgrade when a sizable fraction of the community has !
- Useful to have access to GRID SE (DESY dcache).

Observations about the NAF

- Convenient to access AFS and GRID storage elements.
  - Both Calice and ILC have pre-installed software on AFS $\Rightarrow$
  - Same SW on your desk-top and on the NAF.
  - Corollary: It is preferable that the NAF avoids being on the bleeding edge: Don't be the first to go to the latest-and-greatest SL-version, but upgrade when a sizable fraction of the community has !
- Useful to have access to GRID SE (DESY dcache).

Problems

- Occasional Lustre hick-ups - “image busy”
- AFS issues. A pity that naf.desy.de can’t be just desy.de.
  - Why is it similar, but still different ?
  - naf_token on Mac-OS ? On Ubuntu ? On Windows ?
- Lack of a comprehensive explanation on how job priorities are calculated, and/or a simple view of the queueing status of my jobs.
- The somewhat unclear status of files on Lustre. Temporary, yes, but never gets deleted.
- A potential problem: Self-organising community wrt. interactive use and disk-space. Works fine now, but might degenerate at any time. Hope not.

Problems

- Occasional Lustre hick-ups - “image busy”
- AFS issues. A pity that naf.desy.de can’t be just desy.de.
  - Why is it similar, but still different ?
  - naf_token on Mac-OS ? On Ubuntu ? On Windows ?
- Lack of a comprehensive explanation on how job priorities are calculated, and/or a simple view of the queueing status of my jobs.
- The somewhat unclear status of files on Lustre. Temporary, yes, but never gets deleted.
- A potential problem: Self-organising community wrt. interactive use and disk-space. Works fine now, but might degenerate at any time. Hope not.

Problems

- Occasional Lustre hick-ups - “image busy”
- AFS issues. A pity that naf.desy.de can’t be just desy.de.
  - Why is it similar, but still different ?
  - naf_token on Mac-OS ? On Ubuntu ? On Windows ?
- Lack of a comprehensive explanation on how job priorities are calculated, and/or a simple view of the queueing status of my jobs.
- The somewhat unclear status of files on Lustre. Temporary, yes, but never gets deleted.
- A potential problem: Self-organising community wrt. interactive use and disk-space. Works fine now, but might degenerate at any time. Hope not.

Problems

- Occasional Lustre hick-ups - “image busy”
- AFS issues. A pity that naf.desy.de can’t be just desy.de.
  - Why is it similar, but still different ?
  - naf_token on Mac-OS ? On Ubuntu ? On Windows ?
- Lack of a comprehensive explanation on how job priorities are calculated, and/or a simple view of the queueing status of my jobs.
- The somewhat unclear status of files on Lustre. Temporary, yes, but never gets deleted.
- A potential problem: Self-organising community wrt. interactive use and disk-space. Works fine now, but might degenerate at any time. Hope not.

Problems

- Occasional Lustre hick-ups - “image busy”
- AFS issues. A pity that naf.desy.de can’t be just desy.de.
  - Why is it similar, but still different ?
  - naf_token on Mac-OS ? On Ubuntu ? On Windows ?
- Lack of a comprehensive explanation on how job priorities are calculated, and/or a simple view of the queueing status of my jobs.
- The somewhat unclear status of files on Lustre. Temporary, yes, but never gets deleted.
- A potential problem: Self-organising community wrt. interactive use and disk-space. Works fine now, but might degenerate at any time. Hope not.

Summary

- A few issues where pointed out, but:
- NAF is the most convenient batch system I've ever used $\Rightarrow$

NAF works fine !

Summary

- A few issues where pointed out, but:
- NAF is the most convenient batch system I've ever used $\Rightarrow$

NAF works fine !

Summary

- A few issues where pointed out, but:
- NAF is the most convenient batch system I've ever used $\Rightarrow$

NAF works fine !