



NIDays

WORLDWIDE GRAPHICAL SYSTEM DESIGN
CONFERENCE



Tips and Tricks to Speed LabVIEW Performance

Useful Nuggets to Accelerate Your VIs



Presenter name
NI-Days 2010

Agenda

- Programming Techniques
- Algorithm Selection
- LabVIEW Environment Tools
- New LabVIEW Usability Features
(not really “performance” nuggets, but still fun to talk about)

Programming Techniques

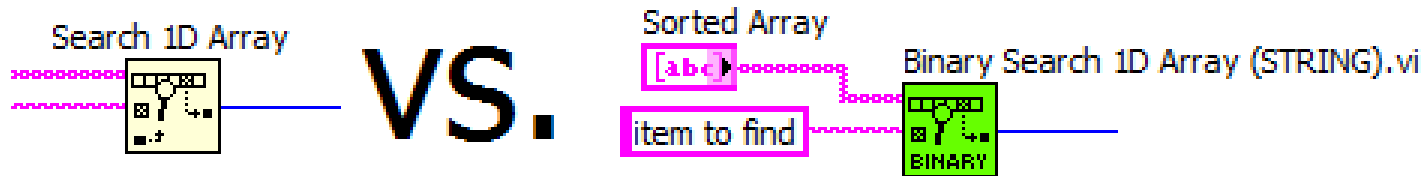
Six specific suggestions on how to improve the performance of seemingly simple operations in LabVIEW

#1. Build Array Ordering



- Appending to the end of an array is much more efficient than prepending to the beginning
- If you need to insert items at the beginning, insert at the end instead and use a single Reverse Array when finished

#2. Binary Search vs. Linear



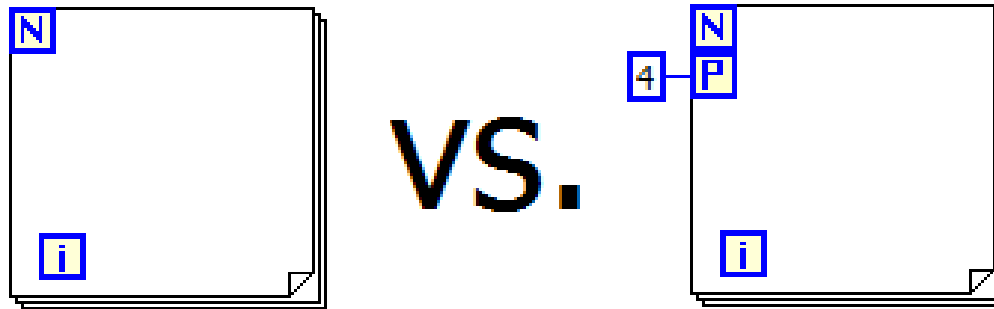
- The Search 1D Array function in LabVIEW does a linear search from the beginning of the array
- If you have a way to keep your array sorted, a binary search can be much faster

#3. Data Type Selection



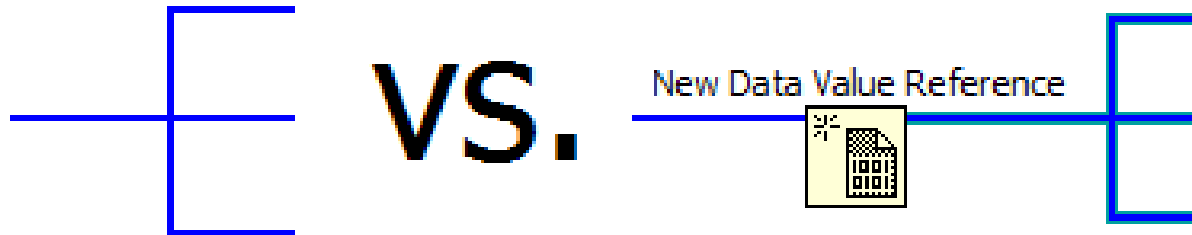
- Processing the same data as different data types can yield the same results, but with varying performance
- Example: LabVIEW core MD5 calculation vs. OpenG MD5 calculation

#4. Parallel For Loop



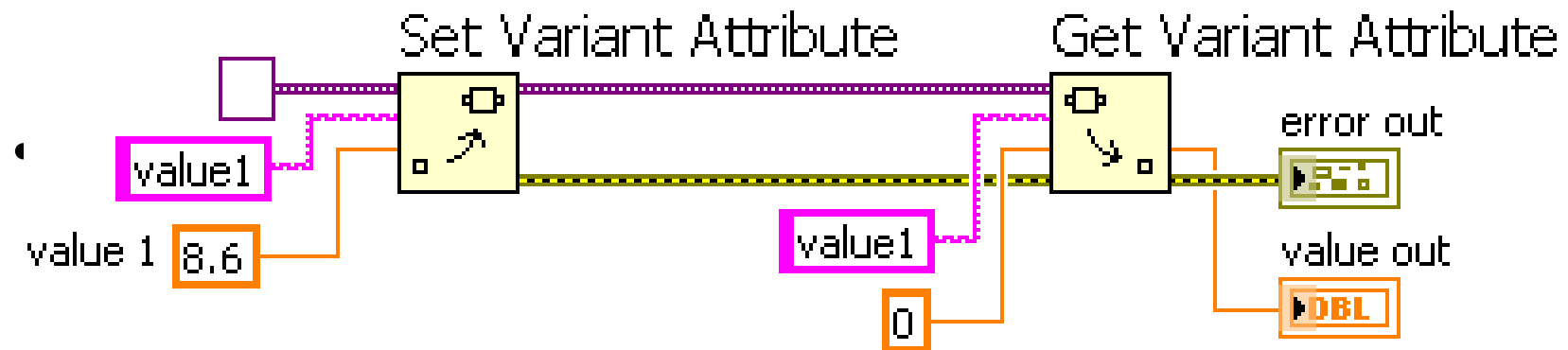
- The Parallel For Loop is a new feature in LabVIEW 2009
- With some For Loops, it enables parallel processing of independent iterations that otherwise would execute sequentially

#5. Data Value References



- Data Value References are also new in LabVIEW 2009
- If you know you don't need to copy array (or other large) data, you can manipulate references to the data instead of the data itself

#6. Variant Attributes for Data Storage



- “Red-black tree” search algorithm used (as opposed to the linear search of Search 1D Array or a While Loop)
- Order $\log(n)$ vs. Order (n) search

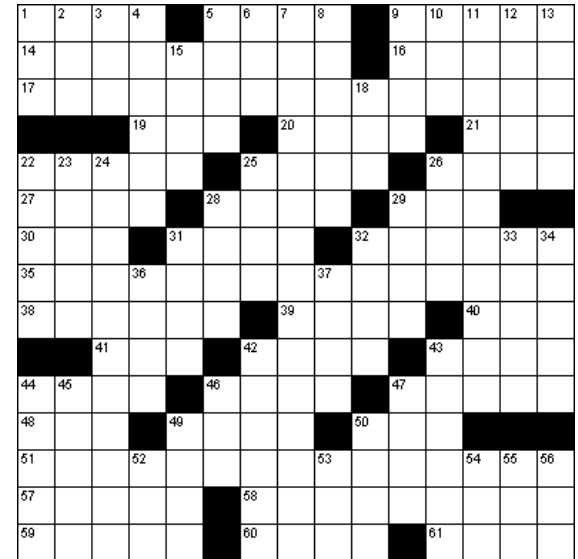
Algorithm Selection

There's an easy way to solve this, but let's try to find the *fastest* solution.

Crossword Clue Helper

- 14 Across – Former U.S. Vice President's Plan for Choosing a New Drummer?

_ L _ _ R I _ _ _



How do we write a VI to generate possible crossword clue solutions?

LabVIEW Environment Tools

These LabVIEW toolkits can help you locate performance problems in your VIs.

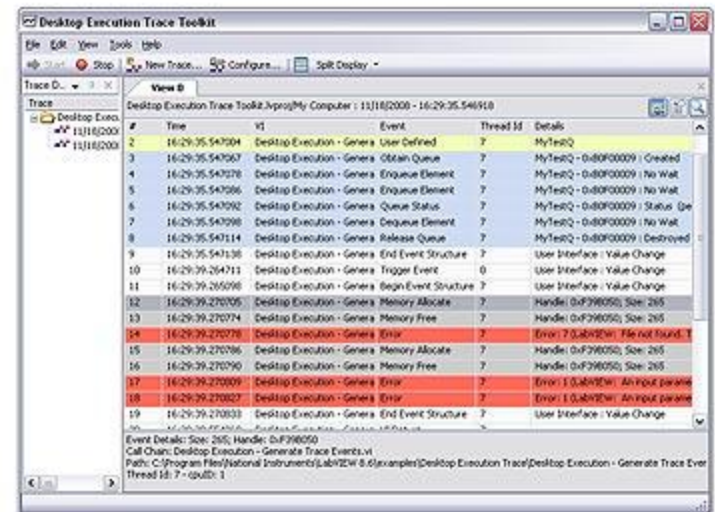
VI Analyzer Toolkit - Performance Tests

☐ ☒ Block Diagram

☐ ☒ Performance

- ☒ Arrays and Strings in Loops
- ☒ Coercion Dots
- ☒ Enabled Debugging
- ☒ Wait in While Loop
- ☒ Wired Terminals in Subdiagrams

Desktop Execution Trace Toolkit

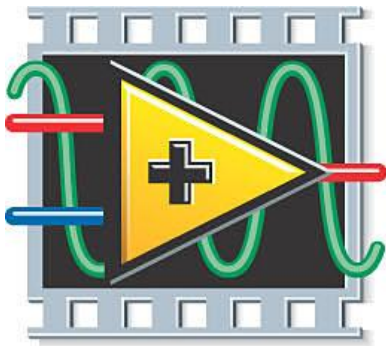


LabVIEW Usability Features

Not exactly “performance” nuggets, but still worth mentioning...

- **Quick Drop Keyboard Shortcuts**
 - currently shipping in LV 2009
- **JKI Right-Click Framework**
 - jkisoft.com/labs
- **Partial Block Diagram Cleanup**
- **Tunnel Auto Wiring**
- **Front Panel Multi-select Properties**
- **LVSpeak**
 - lavag.org – search for ‘LVSpeak’

Thank you for attending!



NATIONAL INSTRUMENTS

LabVIEW™

Check out Darrens' blog: labviewartisan.blogspot.com

Bonus Programming Techniques

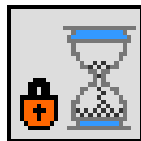
More tips on how to improve the performance of seemingly simple operations in LabVIEW

#7. Defer Panel Updates

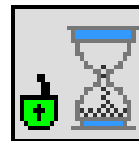


- Useful when you have a front panel update operation that may take a while
- Usually used in conjunction with Set Busy.vi and Unset Busy.vi

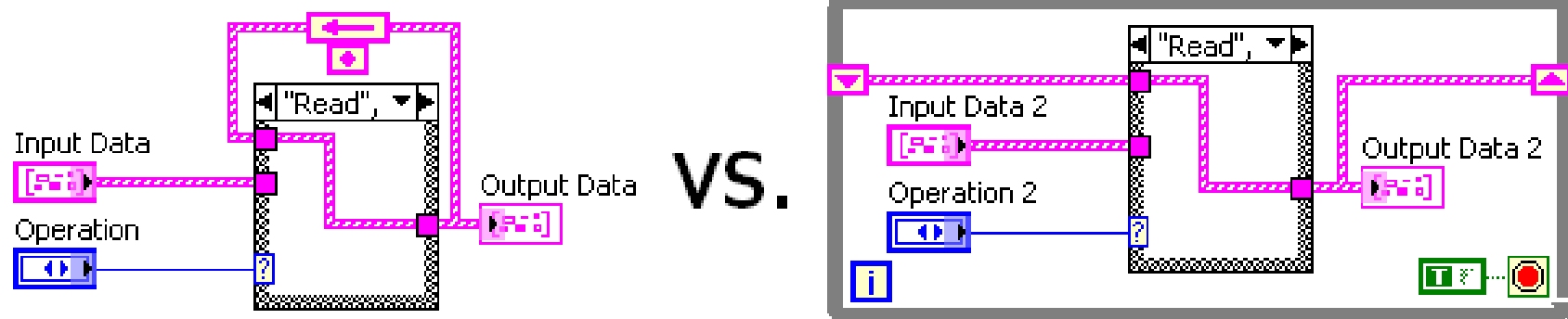
Set Busy.vi



Unset Busy.vi



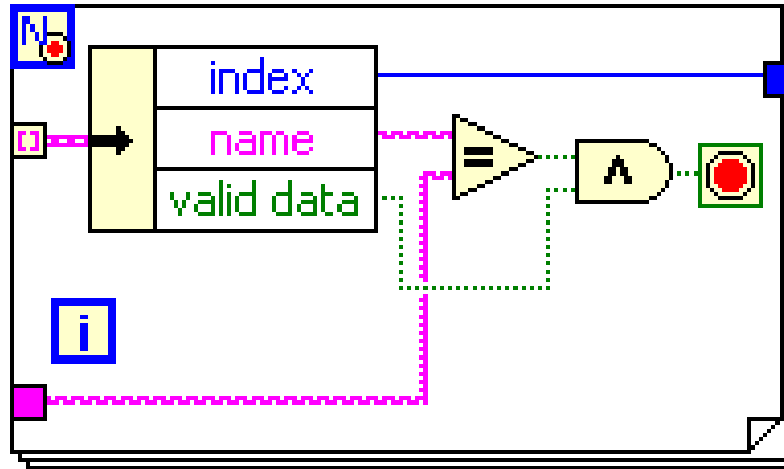
#8. Feedback Nodes



- The loop-free Feedback Node functions identically to a Functional Global (i.e. LV2-style Global)

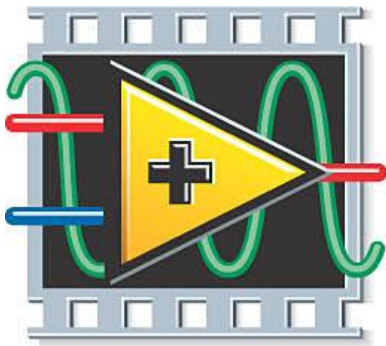
#9. For Loop with Break

For Loop with Break



- Allows for iteration over a data set with the option to break in the middle of operation
- Eliminates the need for extra calculations (While Loop) and extraneous iterations (For Loop without break)

**Well, that's all.
Thank you for attending!**



NATIONAL INSTRUMENTS
LabVIEW™

*To read more tips and tricks,
check out Darrens' blog: labviewartisan.blogspot.com*