Zhao, Jun-Hui <jzhao@cfa.harvard.edu>

The our data's dynamic range problem

7 messages

Young, Ken <kyoung@cfa.harvard.edu> Mon, Apr 3, 2017 at 11:19 AM
To: Sridharan Tirupati Kumara <tksridha@cfa.harvard.edu>, Mark Gurwell <mgurwell@cfa.harvard.edu>, David Wilner <dwilner@cfa.harvard.edu>, Chunhua Qi <cqi@cfa.harvard.edu>, "Zhao, Jun-Hui" <jzhao@cfa.harvard.edu>, Eric Keto <eketo@cfa.harvard.edu>, "Petitpas, Glen" <gpetitpas@cfa.harvard.edu>, Raymond Blundell <rblundell@cfa.harvard.edu>, Ramprasad Rao <rrao@asiaa.sinica.edu.tw>, "Patel, Nimesh" <npatel@cfa.harvard.edu>, Qizhou Zhang <qzhang@cfa.harvard.edu>, Garrett Keating <garrett.keating@cfa.harvard.edu>, "Primiani, Rurik" <rprimiani@cfa.harvard.edu>, Thomas Cooper <tcooper@cfa.harvard.edu>

As you know, there is a problem with our data file format when we have strong RFI spikes in the IF. We inherited our data format from OVRO, and we store the visibilities as 16 bit integers with a scale factor. That means that if an interference spike is more than a few hundred times stronger than the noise level, which is very, very often the case, most of the dynamic range available in a 16 bit integer is used to represent the spike, and the noise, which contains our actual signal of interest, gets expressed using only the few least significant bits of the 16 bit integer. I initially made dataCatcher calculate where the YIG spikes would show up in the spectra, and I clipped them. But we see spikes at the YIG frequencies, the YIG frequencies +- 200 MHz, the gunn frequency, harmonics of the 240 LO frequency, and some frequencies I don't understand at all. So dataCatcher is no longer trying to calculate where RIF will appear - it is just clipping any data above a certain level.

This is very unsatisfying. We should do despiking offline, so that if it is done incorrectly it can be undone. We should not be compromising the quality of our data because of a decision OVRO made back in the 1990s when a 2 GByte disk drive caused people to salivate uncontrollably. Simon made a suggestion about how to address this, during our wideband meeting last Friday. He suggested that the online code be modified so that it stores the visibilities as 16 bit floats, which have a far greater dynamic range. I could write an offline tool that would convert the data set's 16 bit floats into the 16 bit integers that the data reduction packages expect. It would be a program like SMARechunker - it could actually be folded into SMARechunker, if we want to. Since we very often rechunk the data now, adding this conversion to SMARechunker would not add an addition data reduction step, in most cases. We could then do the despiking in SMARechunker, without worrying about screwing up the original data.

What do people think of Simon's idea?

Taco

Tirupati Kumara, Sridharan <tksridharan@cfa.harvard.edu> Mon, Apr 3, 2017 at 11:29 AM
To: "Young, Ken" <kyoung@cfa.harvard.edu>
Cc: Sridharan Tirupati Kumara <tksridha@cfa.harvard.edu>, Mark Gurwell <mgurwell@cfa.harvard.edu>, David Wilner <dwilner@cfa.harvard.edu>, Chunhua Qi <cqi@cfa.harvard.edu>, "Zhao, Jun-Hui" <jzhao@cfa.harvard.edu>, Eric Keto <eketo@cfa.harvard.edu>, "Petitpas, Glen" <gpetitpas@cfa.harvard.edu>, Raymond Blundell <rblundell@cfa.harvard.edu>, Ramprasad Rao <rrao@asiaa.sinica.edu.tw>, "Patel, Nimesh" <npatel@cfa.harvard.edu>, Qizhou Zhang <qzhang@cfa.harvard.edu>, Garrett Keating <garrett.keating@cfa.harvard.edu>, "Primiani, Rurik" <rprimiani@cfa.harvard.edu>, Thomas Cooper <tcooper@cfa.harvard.edu>

Sounds like a good idea to me.

TK.

On Mon, Apr 3, 2017 at 11:19 AM, Young, Ken <kyoung@cfa.harvard.edu> wrote:

> As you know, there is a problem with our data file format when we have strong RFI spikes in the IF. We inherited our data format from OVRO, and we store the visibilities as 16 bit integers with a scale factor. That means that if an interference spike is more than a few hundred times stronger than the noise level, which is very, very often the case, most of the dynamic range available in a 16 bit integer is used to represent the spike, and the noise, which contains our actual signal of interest, gets expressed using only the few least significant bits of the 16 bit integer. I initially made dataCatcher calculate where the YIG spikes would show up in the spectra, and I clipped them. But we see spikes at the YIG frequencies, the YIG frequencies +- 200 MHz, the gunn frequency, harmonics of the 240 LO frequency, and some frequencies I don't understand at all. So dataCatcher is no longer trying to calculate where RIF will appear - it is just clipping any data above a certain level.
>
> This is very unsatisfying. We should do despiking offline, so that if it is done incorrectly it can be undone. We should not be compromising the quality of our data because of a decision OVRO made back in the 1990s when a 2 GByte disk drive caused people to salivate uncontrollably. Simon made a suggestion about how to address this, during our wideband meeting last Friday. He suggested that the online code be modified so that it stores the visibilities as 16 bit floats, which have a far greater dynamic range. I could write an offline tool that would convert the data set's 16 bit floats into the 16 bit integers that the data reduction packages expect. It would be a program like SMARechunker - it could actually be folded into SMARechunker, if we want to. Since we very often rechunk the data now, adding this conversion to SMARechunker would not add an addition data reduction step, in most cases. We could then do the despiking in SMARechunker, without worrying about screwing up the original data.
>
> What do people think of Simon's idea?
>
> Taco

Petitpas, Glen <gpetitpas@cfa.harvard.edu> Mon, Apr 3, 2017 at 11:55 AM
To: "Young, Ken" <kyoung@cfa.harvard.edu>
Cc: Sridharan Tirupati Kumara <tksridha@cfa.harvard.edu>, Mark Gurwell <mgurwell@cfa.harvard.edu>, David Wilner <dwilner@cfa.harvard.edu>, Chunhua Qi <cqi@cfa.harvard.edu>, "Zhao, Jun-Hui" <jzhao@cfa.harvard.edu>, Eric Keto <eketo@cfa.harvard.edu>, Raymond Blundell <rblundell@cfa.harvard.edu>, Ramprasad Rao <rrao@asiaa.sinica.edu.tw>, "Patel, Nimesh" <npatel@cfa.harvard.edu>, Qizhou Zhang <qzhang@cfa.harvard.edu>, Garrett Keating <garrett.keating@cfa.harvard.edu>, "Primiani, Rurik" <rprimiani@cfa.harvard.edu>, Thomas Cooper <tcooper@cfa.harvard.edu>

Hi

I thought the problem with the real time spikes was that they often destroyed the continuum channel, hence the pointing. I missed the details outlined by Simon, but it seems like this would not fix the problem of the spikes corrupting the pointing.

Am I missing something?

Cheers
Glen

On Mon, Apr 3, 2017 at 11:19 AM, Young, Ken <kyoung@cfa.harvard.edu> wrote:
> As you know, there is a problem with our data file format when we have
> strong RFI spikes in the IF. We inherited our data format from OVRO, and
> we store the visibilities as 16 bit integers with a scale factor. That
> means that if an interference spike is more than a few hundred times
> stronger than the noise level, which is very, very often the case, most of
> the dynamic range available in a 16 bit integer is used to represent the
> spike, and the noise, which contains our actual signal of interest, gets
> expressed using only the few least significant bits of the 16 bit integer.
> I initially made dataCatcher calculate where the YIG spikes would show up in
> the spectra, and I clipped them. But we see spikes at the YIG frequencies,
> the YIG frequencies +- 200 MHz, the gunn frequency, harmonics of the 240 LO
> frequency, and some frequencies I don't understand at all. So dataCatcher
> is no longer trying to calculate where RIF will appear - it is just clipping
> any data above a certain level.
>
> This is very unsatisfying. We should do despiking offline, so that if it
> is done incorrectly it can be undone. We should not be compromising the
> quality of our data because of a decision OVRO made back in the 1990s when a
> 2 GByte disk drive caused people to salivate uncontrollably. Simon made a
> suggestion about how to address this, during our wideband meeting last
> Friday. He suggested that the online code be modified so that it stores
> the visibilities as 16 bit floats, which have a far greater dynamic range.
> I could write an offline tool that would convert the data set's 16 bit
> floats into the 16 bit integers that the data reduction packages expect.
> It would be a program like SMARechunker - it could actually be folded into
> SMARechunker, if we want to. Since we very often rechunk the data now,
> adding this conversion to SMARechunker would not add an addition data
> reduction step, in most cases. We could then do the despiking in
> SMARechunker, without worrying about screwing up the original data.
>
> What do people think of Simon's idea?
>
> Taco
>

Mark Gurwell <mgurwell@cfa.harvard.edu> Mon, Apr 3, 2017 at 11:55 AM
To: "Young, Ken" <kyoung@cfa.harvard.edu>
Cc: Sridharan Tirupati Kumara <tksridha@cfa.harvard.edu>, David Wilner <dwilner@cfa.harvard.edu>, Chunhua Qi <cqi@cfa.harvard.edu>, "Zhao, Jun-Hui" <jzhao@cfa.harvard.edu>, Eric Keto <eketo@cfa.harvard.edu>, "Petitpas, Glen" <gpetitpas@cfa.harvard.edu>, Raymond Blundell <rblundell@cfa.harvard.edu>, Ramprasad Rao <rrao@asiaa.sinica.edu.tw>, "Patel, Nimesh" <npatel@cfa.harvard.edu>, Qizhou Zhang <qzhang@cfa.harvard.edu>, Garrett Keating <garrett.keating@cfa.harvard.edu>, "Primiani, Rurik" <rprimiani@cfa.harvard.edu>, Thomas Cooper <tcooper@cfa.harvard.edu>, Holly Sarah Thomas <holly.thomas@cfa.harvard.edu>

Hi Taco, (added Holly to this)

While 'unstatisfying', is there any indication that we are being hurt by the current despike solution?

If we go to 16-bit float, the raw data will be (?)x larger. How big an issue is that. Assuming it isn't a dealbreaker, I am wondering, though, if we should just not have the data reduction use the 16-bit float, rather

than store the data, then require a conversion.

Charlie, how tremendously bad would it be to read in floats for the visibility data vs integers? What is the effect on speed, and what about the stored MIR-format files? What effort would be involved in converting MIR to handle such data?

Jun-Hui, same questions but for Miriad?

Mark

On Mon, Apr 3, 2017 at 11:19 AM, Young, Ken <kyoung@cfa.harvard.edu> wrote:

> As you know, there is a problem with our data file format when we have strong RFI spikes in the IF. We inherited our data format from OVRO, and we store the visibilities as 16 bit integers with a scale factor. That means that if an interference spike is more than a few hundred times stronger than the noise level, which is very, very often the case, most of the dynamic range available in a 16 bit integer is used to represent the spike, and the noise, which contains our actual signal of interest, gets expressed using only the few least significant bits of the 16 bit integer. I initially made dataCatcher calculate where the YIG spikes would show up in the spectra, and I clipped them. But we see spikes at the YIG frequencies, the YIG frequencies +- 200 MHz, the gunn frequency, harmonics of the 240 LO frequency, and some frequencies I don't understand at all. So dataCatcher is no longer trying to calculate where RIF will appear - it is just clipping any data above a certain level.
>
> This is very unsatisfying. We should do despiking offline, so that if it is done incorrectly it can be undone. We should not be compromising the quality of our data because of a decision OVRO made back in the 1990s when a 2 GByte disk drive caused people to salivate uncontrollably. Simon made a suggestion about how to address this, during our wideband meeting last Friday. He suggested that the online code be modified so that it stores the visibilities as 16 bit floats, which have a far greater dynamic range. I could write an offline tool that would convert the data set's 16 bit floats into the 16 bit integers that the data reduction packages expect. It would be a program like SMARechunker - it could actually be folded into SMARechunker, if we want to. Since we very often rechunk the data now, adding this conversion to SMARechunker would not add an addition data reduction step, in most cases. We could then do the despiking in SMARechunker, without worrying about screwing up the original data.
>
> What do people think of Simon's idea?
>
> Taco

Garrett 'Karto' Keating <garrett.keating@cfa.harvard.edu> Mon, Apr 3, 2017 at 12:01 PM
To: "Petitpas, Glen" <gpetitpas@cfa.harvard.edu>
Cc: "Young, Ken" <kyoung@cfa.harvard.edu>, Sridharan Tirupati Kumara <tksridha@cfa.harvard.edu>, Mark Gurwell <mgurwell@cfa.harvard.edu>, David Wilner <dwilner@cfa.harvard.edu>, Chunhua Qi <cqi@cfa.harvard.edu>, "Zhao, Jun-Hui" <jzhao@cfa.harvard.edu>, Eric Keto <eketo@cfa.harvard.edu>, Raymond Blundell <rblundell@cfa.harvard.edu>, Ramprasad Rao <rrao@asiaa.sinica.edu.tw>, "Patel, Nimesh" <npatel@cfa.harvard.edu>, Qizhou Zhang <qzhang@cfa.harvard.edu>, "Primiani, Rurik" <rprimiani@cfa.harvard.edu>, Thomas Cooper <tcooper@cfa.harvard.edu>

Hey Glen,

That's a separate (but related) problem. What Taco is mentioning here is that because of the way we store the data, a strong tone in the data can lead to data in other channels being effectively zeroed out, or otherwise

losing so many bits that there's an added quantization noise penalty. Two other things that might be worth noting:

1) Half-precsion (i.e., 16-bit) floats yield 11 significant bits, which is just 1 bit shy of what we get under ideal circumstances with the current visibility storage scheme, so there's effectively no additional quantization losses. They are same size as what we currently store (16-bit ints).

2) Because of the decreased exponent range, the smallest number we can store with this scheme is ~10^-4, which may be at the threshold of what we get on blank sky. BUT, we're not likely to need the the exponent range that allows for numbers > 1 (the max half is 65504), so perhaps we can use the current common exponent bits (i.e., "scaleExp") to rescale the range to run from ~2e-9 -> 2.

-Karto

> On Apr 3, 2017, at 11:55 AM, Petitpas, Glen <gpetitpas@cfa.harvard.edu> wrote:
>
> Hi
>
> I thought the problem with the real time spikes was that they often
> destroyed the continuum channel, hence the pointing. I missed the
> details outlined by Simon, but it seems like this would not fix the
> problem of the spikes corrupting the pointing.
>
> Am I missing something?
>
> Cheers
> Glen
>
> On Mon, Apr 3, 2017 at 11:19 AM, Young, Ken <kyoung@cfa.harvard.edu> wrote:
>> As you know, there is a problem with our data file format when we have
>> strong RFI spikes in the IF. We inherited our data format from OVRO, and
>> we store the visibilities as 16 bit integers with a scale factor. That
>> means that if an interference spike is more than a few hundred times
>> stronger than the noise level, which is very, very often the case, most of
>> the dynamic range available in a 16 bit integer is used to represent the
>> spike, and the noise, which contains our actual signal of interest, gets
>> expressed using only the few least significant bits of the 16 bit integer.
>> I initially made dataCatcher calculate where the YIG spikes would show up in
>> the spectra, and I clipped them. But we see spikes at the YIG frequencies,
>> the YIG frequencies +- 200 MHz, the gunn frequency, harmonics of the 240 LO
>> frequency, and some frequencies I don't understand at all. So dataCatcher
>> is no longer trying to calculate where RIF will appear - it is just clipping
>> any data above a certain level.
>>
>> This is very unsatisfying. We should do despiking offline, so that if it
>> is done incorrectly it can be undone. We should not be compromising the
>> quality of our data because of a decision OVRO made back in the 1990s when a
>> 2 GByte disk drive caused people to salivate uncontrollably. Simon made a
>> suggestion about how to address this, during our wideband meeting last
>> Friday. He suggested that the online code be modified so that it stores
>> the visibilities as 16 bit floats, which have a far greater dynamic range.
>> I could write an offline tool that would convert the data set's 16 bit
>> floats into the 16 bit integers that the data reduction packages expect.

>> It would be a program like SMARechunker - it could actually be folded into
>> SMARechunker, if we want to. Since we very often rechunk the data now,
>> adding this conversion to SMARechunker would not add an addition data
>> reduction step, in most cases. We could then do the despiking in
>> SMARechunker, without worrying about screwing up the original data.
>>
>> What do people think of Simon's idea?
>>
>> Taco
>>

Qi, Chunhua <cqi@cfa.harvard.edu> Mon, Apr 3, 2017 at 12:07 PM
To: Mark Gurwell <mgurwell@cfa.harvard.edu>
Cc: "Young, Ken" <kyoung@cfa.harvard.edu>, Sridharan Tirupati Kumara <tksridha@cfa.harvard.edu>, David Wilner <dwilner@cfa.harvard.edu>, "Zhao, Jun-Hui" <jzhao@cfa.harvard.edu>, Eric Keto <eketo@cfa.harvard.edu>, "Petitpas, Glen" <gpetitpas@cfa.harvard.edu>, Raymond Blundell <rblundell@cfa.harvard.edu>, Ramprasad Rao <rrao@asiaa.sinica.edu.tw>, "Patel, Nimesh" <npatel@cfa.harvard.edu>, Qizhou Zhang <qzhang@cfa.harvard.edu>, Garrett Keating <garrett.keating@cfa.harvard.edu>, "Primiani, Rurik" <rprimiani@cfa.harvard.edu>, Thomas Cooper <tcooper@cfa.harvard.edu>, Holly Sarah Thomas <holly.thomas@cfa.harvard.edu>

I actually don't know how bad it would be to read in floats. We can try on a test data.

Charlie

On Mon, Apr 3, 2017 at 11:55 AM, Mark Gurwell <mgurwell@cfa.harvard.edu> wrote:

> Hi Taco, (added Holly to this)
>
> While 'unstatisfying', is there any indication that we are being hurt by the current despike solution?
>
> If we go to 16-bit float, the raw data will be (?)x larger. How big an issue is that. Assuming it isn't a dealbreaker, I am wondering, though, if we should just not have the data reduction use the 16-bit float, rather than store the data, then require a conversion.
>
> Charlie, how tremendously bad would it be to read in floats for the visibility data vs integers? What is the effect on speed, and what about the stored MIR-format files? What effort would be involved in converting MIR to handle such data?
>
> Jun-Hui, same questions but for Miriad?
>
> Mark
>
> On Mon, Apr 3, 2017 at 11:19 AM, Young, Ken <kyoung@cfa.harvard.edu> wrote:
>
>> As you know, there is a problem with our data file format when we have strong RFI spikes in the IF. We inherited our data format from OVRO, and we store the visibilities as 16 bit integers with a scale factor. That means that if an interference spike is more than a few hundred times stronger than the noise level, which is very, very often the case, most of the dynamic range available in a 16 bit integer is used to represent the spike, and the noise, which contains our actual signal of interest, gets expressed using only the few least significant bits of the 16 bit integer. I initially made dataCatcher calculate where the YIG spikes would show up in the spectra, and I clipped them. But we see spikes at the YIG frequencies, the YIG frequencies +- 200 MHz, the gunn frequency, harmonics of the 240 LO frequency, and some

> frequencies I don't understand at all. So dataCatcher is no longer trying to calculate where RIF will appear - it is just clipping any data above a certain level.
>
> This is very unsatisfying. We should do despiking offline, so that if it is done incorrectly it can be undone. We should not be compromising the quality of our data because of a decision OVRO made back in the 1990s when a 2 GByte disk drive caused people to salivate uncontrollably. Simon made a suggestion about how to address this, during our wideband meeting last Friday. He suggested that the online code be modified so that it stores the visibilities as 16 bit floats, which have a far greater dynamic range. I could write an offline tool that would convert the data set's 16 bit floats into the 16 bit integers that the data reduction packages expect. It would be a program like SMARechunker - it could actually be folded into SMARechunker, if we want to. Since we very often rechunk the data now, adding this conversion to SMARechunker would not add an addition data reduction step, in most cases. We could then do the despiking in SMARechunker, without worrying about screwing up the original data.
>
> What do people think of Simon's idea?
>
> Taco

Petitpas, Glen <gpetitpas@cfa.harvard.edu> Mon, Apr 3, 2017 at 12:09 PM
To: Garrett 'Karto' Keating <garrett.keating@cfa.harvard.edu>
Cc: "Young, Ken" <kyoung@cfa.harvard.edu>, Sridharan Tirupati Kumara <tksridha@cfa.harvard.edu>, Mark Gurwell <mgurwell@cfa.harvard.edu>, David Wilner <dwilner@cfa.harvard.edu>, Chunhua Qi <cqi@cfa.harvard.edu>, "Zhao, Jun-Hui" <jzhao@cfa.harvard.edu>, Eric Keto <eketo@cfa.harvard.edu>, Raymond Blundell <rblundell@cfa.harvard.edu>, Ramprasad Rao <rrao@asiaa.sinica.edu.tw>, "Patel, Nimesh" <npatel@cfa.harvard.edu>, Qizhou Zhang <qzhang@cfa.harvard.edu>, "Primiani, Rurik" <rprimiani@cfa.harvard.edu>, Thomas Cooper <tcooper@cfa.harvard.edu>

> Yes, but I'm advocating in favour of the real-time de-spiker for
> operational/i-pointing reasons.
>
> Loosing dynamic range is bad, but allowing us to safely have *bigger*
> spikes will only make the fact that we cannot point when they occur a
> bigger problem.
>
> G
>
> On Mon, Apr 3, 2017 at 12:01 PM, Garrett 'Karto' Keating
> <garrett.keating@cfa.harvard.edu> wrote:
> > Hey Glen,
> >
> > That’s a separate (but related) problem. What Taco is mentioning here is that because of the way we store the data, a strong tone in the data can lead to data in other channels being effectively zeroed out, or otherwise losing so many bits that there’s an added quantization noise penalty. Two other things that might be worth noting:
> >
> > 1) Half-precsion (i.e., 16-bit) floats yield 11 significant bits, which is just 1 bit shy of what we get under ideal circumstances with the current visibility storage scheme, so there’s effectively no additional quantization losses. They are same size as what we currently store (16-bit ints).
> >
> > 2) Because of the decreased exponent range, the smallest number we can store with this scheme is

~10^-4, which may be at the threshold of what we get on blank sky. BUT, we're not likely to need the the exponent range that allows for numbers > 1 (the max half is 65504), so perhaps we can use the current common exponent bits (i.e., "scaleExp") to rescale the range to run from ~2e-9 -> 2.
>
> -Karto
>
>> On Apr 3, 2017, at 11:55 AM, Petitpas, Glen <gpetitpas@cfa.harvard.edu> wrote:
>>
>> Hi
>>
>> I thought the problem with the real time spikes was that they often
>> destroyed the continuum channel, hence the pointing. I missed the
>> details outlined by Simon, but it seems like this would not fix the
>> problem of the spikes corrupting the pointing.
>>
>> Am I missing something?
>>
>> Cheers
>> Glen
>>
>> On Mon, Apr 3, 2017 at 11:19 AM, Young, Ken <kyoung@cfa.harvard.edu> wrote:
>>> As you know, there is a problem with our data file format when we have
>>> strong RFI spikes in the IF. We inherited our data format from OVRO, and
>>> we store the visibilities as 16 bit integers with a scale factor. That
>>> means that if an interference spike is more than a few hundred times
>>> stronger than the noise level, which is very, very often the case, most of
>>> the dynamic range available in a 16 bit integer is used to represent the
>>> spike, and the noise, which contains our actual signal of interest, gets
>>> expressed using only the few least significant bits of the 16 bit integer.
>>> I initially made dataCatcher calculate where the YIG spikes would show up in
>>> the spectra, and I clipped them. But we see spikes at the YIG frequencies,
>>> the YIG frequencies +- 200 MHz, the gunn frequency, harmonics of the 240 LO
>>> frequency, and some frequencies I don't understand at all. So dataCatcher
>>> is no longer trying to calculate where RIF will appear - it is just clipping
>>> any data above a certain level.
>>>
>>> This is very unsatisfying. We should do despiking offline, so that if it
>>> is done incorrectly it can be undone. We should not be compromising the
>>> quality of our data because of a decision OVRO made back in the 1990s when a
>>> 2 GByte disk drive caused people to salivate uncontrollably. Simon made a
>>> suggestion about how to address this, during our wideband meeting last
>>> Friday. He suggested that the online code be modified so that it stores
>>> the visibilities as 16 bit floats, which have a far greater dynamic range.
>>> I could write an offline tool that would convert the data set's 16 bit
>>> floats into the 16 bit integers that the data reduction packages expect.
>>> It would be a program like SMARechunker - it could actually be folded into
>>> SMARechunker, if we want to. Since we very often rechunk the data now,
>>> adding this conversion to SMARechunker would not add an addition data
>>> reduction step, in most cases. We could then do the despiking in
>>> SMARechunker, without worrying about screwing up the original data.
>>>
>>> What do people think of Simon's idea?
>>>

```
>>> Taco
>>>
>
```