

Arkouda Boolean pdarray Indexing

mimics NumPy boolean indexing

Algorithm/Code Comments

- simple parallel-prefix/scan-based algorithm
- size and bounds checks
- the real code has some optimizations like refs and aggregation
- chapel code in `src/IndexingMsg.chpl`

Boolean Gather – Compression Indexing

$$Y = X[\text{truth}]$$

i	0	1	2	3	4	5	6	7	8	9
X	1	2	5	5	1	5	2	5	3	1
truth = (X == 5)	f	f	t	t	f	t	f	t	f	f

iv = + scan truth	0	0	1	2	2	3	3	4	4	4
if truth then iv - 1			0	1		2		3		
Y = X[truth]	5	5	5	5						

Compression Indexing Code

```
#Python code  
truth = X == 5  
Y = X[truth]
```

```
//Chapel code  
iv = + scan truth;  
[i in X.domain] if (truth[i] == true) {Y[iv[i]-1] = X[i];}  
// iv[i]-1 for zero base index
```


Boolean Scatter – Expansion Indexing

$X[\text{truth}] = Y$

i	0	1	2	3	4	5	6	7	8	9
Y	5	5	5	5						
X	1	2	-1	-1	1	-1	2	-1	3	1
truth	f	f	t	t	f	t	f	t	f	f

iv = + scan truth	0	0	1	2	2	3	3	4	4	4
if truth then iv - 1			0	1		2		3		
$X[\text{truth}] = Y$	1	2	5	5	1	5	2	5	3	1

Expansion Indexing Code

```
#Python code  
X[truth] = Y
```

```
//Chapel code  
iv = + scan truth;  
[i in X.domain] if (truth[i] == true) {X[i] = Y[iv[i]-1];}  
// iv[i]-1 for zero base index
```


look at code now