00:00:00:04 - 00:00:01:23
CETRA’s approach to transit detection

00:00:01:23 - 00:00:04:00
is structured in two stages.

00:00:04:00 - 00:00:05:16
The first is a linear search.

00:00:05:16 - 00:00:07:19
It sweeps through the light curve in time,

00:00:07:19 - 00:00:09:16
evaluating the likelihood of a transit

00:00:09:16 - 00:00:11:03
at each point on a grid

00:00:11:03 - 00:00:13:18
of transit times and durations.

00:00:13:18 - 00:00:16:01
The second is a periodic search.

00:00:16:01 - 00:00:18:01
It takes the output of the linear search

00:00:18:01 - 00:00:21:02
and phase-folds it over a grid of trial periods

00:00:21:02 - 00:00:24:03
to find the most likely periodic signal.

00:00:24:03 - 00:00:26:11
This linear, then periodic structure is

00:00:26:11 - 00:00:27:13
a deliberate departure

00:00:27:13 - 00:00:29:20
from how BLS and TLS work,

00:00:29:20 - 00:00:31:08

and it's motivated directly

00:00:31:08 - 00:00:33:09
by the GPU context.

00:00:33:09 - 00:00:36:04
BLS and TLS both phase-fold the light

00:00:36:04 - 00:00:36:24
curve first,

00:00:36:24 - 00:00:38:07
then search for transit

00:00:38:07 - 00:00:40:05
signals in the folded data.

00:00:40:05 - 00:00:40:24
Phase-folding

00:00:40:24 - 00:00:42:07
requires sorting -

00:00:42:07 - 00:00:43:11
reordering the data points

00:00:43:11 - 00:00:44:19
by their phase -

00:00:44:19 - 00:00:46:00
and as we've just seen,

00:00:46:00 - 00:00:49:00
sorting is something GPUs do poorly.

00:00:49:00 - 00:00:50:22
By searching in linear time first

00:00:50:22 - 00:00:52:08
and only phase-folding

00:00:52:08 - 00:00:54:05
the compact results afterwards,

00:00:54:05 - 00:00:57:13
CETRA avoids this expensive step entirely.

00:00:57:13 - 00:00:58:17
There's a bonus too:

00:00:58:17 - 00:01:00:10
because the linear search produces

00:01:00:10 - 00:01:00:19
a result

00:01:00:19 - 00:01:02:08
for every position in time,

00:01:02:08 - 00:01:04:23
independently of any assumed period,

00:01:04:23 - 00:01:06:22
you get monotransit detection -

00:01:06:22 - 00:01:08:12
the ability to find single,

00:01:08:12 - 00:01:10:03
isolated transit events -

00:01:10:03 - 00:01:11:24
essentially for free.

00:01:11:24 - 00:01:13:02
Long-period planets

00:01:13:02 - 00:01:14:15
will appear as monotransits

00:01:14:15 - 00:01:15:24
until enough of the orbit

00:01:15:24 - 00:01:17:19
is covered by observations

00:01:17:19 - 00:01:19:16
to confirm periodicity,

00:01:19:16 - 00:01:21:10
and being able to flag them early

00:01:21:10 - 00:01:23:05
is scientifically valuable.

00:01:23:05 - 00:01:24:14
The first thing CETRA does

00:01:24:14 - 00:01:26:14
before any searching is resample

00:01:26:14 - 00:01:27:14
the input light curve

00:01:27:14 - 00:01:30:02
onto a regular time grid.

00:01:30:02 - 00:01:31:07
Real light curves

00:01:31:07 - 00:01:33:00
are irregularly sampled.

00:01:33:00 - 00:01:34:10
Even a nominally regular

00:01:34:10 - 00:01:36:01
cadence becomes irregular

00:01:36:01 - 00:01:38:16
once you convert to a common time standard,

00:01:38:16 - 00:01:39:18
and data gaps -

00:01:39:18 - 00:01:41:06
between observing sectors,

00:01:41:06 - 00:01:42:07

during downlinks,

00:01:42:07 - 00:01:42:23
or wherever

00:01:42:23 - 00:01:45:04
observations were simply not taken -

00:01:45:04 - 00:01:47:13
leave holes of arbitrary length.

00:01:47:13 - 00:01:49:13
Classical algorithms like TLS

00:01:49:13 - 00:01:50:06
deal with this

00:01:50:06 - 00:01:52:14
by sorting the data in phase space,

00:01:52:14 - 00:01:54:18
which accommodates irregular sampling

00:01:54:18 - 00:01:55:18
but is expensive

00:01:55:18 - 00:01:59:16
and as we've seen, GPU-unfriendly.

00:01:59:16 - 00:02:01:21
CETRA instead resamples the

00:02:01:21 - 00:02:04:02
light curve to a uniform cadence.

00:02:04:02 - 00:02:06:02
Each input observation is assigned

00:02:06:02 - 00:02:08:00
to the nearest grid point.

00:02:08:00 - 00:02:09:09
Where multiple observations

00:02:09:09 - 00:02:10:17
fall on the same grid point

00:02:10:17 - 00:02:11:11
they are combined

00:02:11:11 - 00:02:13:23
using an inverse-variance weighted mean,

00:02:13:23 - 00:02:15:18
and their uncertainties are propagated

00:02:15:18 - 00:02:17:06
formally.

00:02:17:06 - 00:02:19:06
Grid points with no associated

00:02:19:06 - 00:02:20:16
observations are assigned

00:02:20:16 - 00:02:21:22
null flux values

00:02:21:22 - 00:02:23:18
and infinite uncertainties,

00:02:23:18 - 00:02:24:18
meaning they contribute

00:02:24:18 - 00:02:27:03
nothing to any subsequent calculation,

00:02:27:03 - 00:02:28:17
as though they don't exist.

00:02:28:17 - 00:02:29:22
This might sound like it

00:02:29:22 - 00:02:32:00
throws away information, but it doesn't -

00:02:32:00 - 00:02:33:23
it just reorganises it.

00:02:33:23 - 00:02:36:09
It's worth noting that the resampled grid

00:02:36:09 - 00:02:37:18
can be at a higher cadence

00:02:37:18 - 00:02:39:09
than the input observations,

00:02:39:09 - 00:02:40:04
which limits

00:02:40:04 - 00:02:41:19
any distortion introduced

00:02:41:19 - 00:02:43:21
by the nearest-neighbour assignment.

00:02:43:21 - 00:02:45:14
And because the GPU handles

00:02:45:14 - 00:02:47:16
the extra points so cheaply,

00:02:47:16 - 00:02:50:19
the additional cost is negligible.

00:02:50:19 - 00:02:52:12
The payoff is significant:

00:02:52:12 - 00:02:53:10
once the light curve

00:02:53:10 - 00:02:54:20
is on a regular grid,

00:02:54:20 - 00:02:56:12
finding which data points

00:02:56:12 - 00:02:58:15

fall within any given transit window

00:02:58:15 - 00:03:00:18
becomes trivial arithmetic.

00:03:00:18 - 00:03:03:03
If a transit starts at time t

00:03:03:03 - 00:03:04:17
and has duration d,

00:03:04:17 - 00:03:06:08
the indices of the resampled

00:03:06:08 - 00:03:07:18
light curve elements that fall within

00:03:07:18 - 00:03:08:08
that window

00:03:08:08 - 00:03:09:24
are simply the integer range

00:03:09:24 - 00:03:10:22
from t

00:03:10:22 - 00:03:12:22
divided by the cadence to t

00:03:12:22 - 00:03:15:04
plus d divided by the cadence.

00:03:15:04 - 00:03:16:23
No searching, no sorting -

00:03:16:23 - 00:03:18:19
just arithmetic.

00:03:18:19 - 00:03:19:24
This makes the inner

00:03:19:24 - 00:03:21:12
loop of the linear search

00:03:21:12 - 00:03:22:23
kernel extremely cheap,

00:03:22:23 - 00:03:24:08
and it's one of the most important

00:03:24:08 - 00:03:25:06
design decisions

00:03:25:06 - 00:03:26:12
in CETRA.

00:03:26:12 - 00:03:28:19
It also means that data gaps are handled

00:03:28:19 - 00:03:30:11
correctly and formally.

00:03:30:11 - 00:03:31:21
A null point inside

00:03:31:21 - 00:03:33:08
a transit window contributes

00:03:33:08 - 00:03:35:18
nothing to the likelihood calculation.

00:03:35:18 - 00:03:38:12
TLS, by contrast, assumes uniform

00:03:38:12 - 00:03:40:04
phase-space sampling,

00:03:40:04 - 00:03:41:04
which breaks down

00:03:41:04 - 00:03:43:02
in the presence of gaps,

00:03:43:02 - 00:03:44:20
particularly for long-period planets

00:03:44:20 - 00:03:46:09
with few observed transits

00:03:46:09 - 00:03:47:12
where a gap falling on

00:03:47:12 - 00:03:49:09
one transit can meaningfully bias

00:03:49:09 - 00:03:50:01
the result.

00:03:50:01 - 00:03:52:19
CETRA uses a physically motivated exoplanet

00:03:52:19 - 00:03:55:03
transit model rather than a box shape.

00:03:55:03 - 00:03:57:18
The model is stored as a 1D array of flux

00:03:57:18 - 00:03:59:21
values sampled at regular intervals

00:03:59:21 - 00:04:01:12
in time from the start

00:04:01:12 - 00:04:03:04
to the end of the transit.

00:04:03:04 - 00:04:04:13
When the linear search needs

00:04:04:13 - 00:04:04:23
to know

00:04:04:23 - 00:04:06:22
what the model flux is at a given point

00:04:06:22 - 00:04:08:17
inside a transit window,

00:04:08:17 - 00:04:11:06

it uses nearest-neighbour interpolation:

00:04:11:06 - 00:04:13:03
it simply looks up the array element

00:04:13:03 - 00:04:14:12
closest to the relevant

00:04:14:12 - 00:04:15:12
fractional position

00:04:15:12 - 00:04:17:09
within the transit duration.

00:04:17:09 - 00:04:20:05
This lookup is O(1) - constant time

00:04:20:05 - 00:04:22:01
regardless of model size.

00:04:22:01 - 00:04:24:11
With 1024 model elements,

00:04:24:11 - 00:04:25:17
which is the default,

00:04:25:17 - 00:04:26:24
the maximum interpolation

00:04:26:24 - 00:04:27:18
error is around

00:04:27:18 - 00:04:30:23
1% and the mean error is around 0.1%,

00:04:30:23 - 00:04:32:07
which is entirely acceptable

00:04:32:07 - 00:04:34:16
for transit detection purposes.

00:04:34:16 - 00:04:35:19
The transit model is loaded

00:04:35:19 - 00:04:36:19
into shared memory

00:04:36:19 - 00:04:38:01
at the start of each thread

00:04:38:01 - 00:04:39:11
block's execution.

00:04:39:11 - 00:04:41:06
This is a key optimisation.

00:04:41:06 - 00:04:42:02
The linear search

00:04:42:02 - 00:04:43:01
kernel compares

00:04:43:01 - 00:04:44:05
the model to the light curve

00:04:44:05 - 00:04:46:10
thousands of times per thread block -

00:04:46:10 - 00:04:47:15
once for each data

00:04:47:15 - 00:04:48:20
point in each transit

00:04:48:20 - 00:04:50:12
window it evaluates.

00:04:50:12 - 00:04:52:14
If the model lived in global memory,

00:04:52:14 - 00:04:53:23
each of those comparisons

00:04:53:23 - 00:04:56:24
would incur a slow global memory read.

00:04:56:24 - 00:04:58:16
By loading it into shared memory

00:04:58:16 - 00:04:59:24
once at the start,

00:04:59:24 - 00:05:02:07
all subsequent lookups are fast.

00:05:02:07 - 00:05:03:01
In practice,

00:05:03:01 - 00:05:04:21
this roughly halved the execution

00:05:04:21 - 00:05:06:24
time of the linear search kernel.

00:05:06:24 - 00:05:07:19
One optimisation

00:05:07:19 - 00:05:09:16
that had a surprisingly large impact,

00:05:09:16 - 00:05:11:17
particularly on consumer and workstation

00:05:11:17 - 00:05:13:20
hardware, was switching from 64-bit

00:05:13:20 - 00:05:15:11
to 32-bit floating

00:05:15:11 - 00:05:17:00
point arithmetic throughout.

00:05:17:00 - 00:05:18:20
Most consumer and workstation

00:05:18:20 - 00:05:20:20
GPUs have significantly lower

00:05:20:20 - 00:05:23:01

throughput for 64-bit operations

00:05:23:01 - 00:05:24:20
than for 32-bit.

00:05:24:20 - 00:05:27:02
On my desktop GPU, switching CETRA

00:05:27:02 - 00:05:28:03
from 64-bit

00:05:28:03 - 00:05:29:01
to 32-bit

00:05:29:01 - 00:05:32:06
reduced the run time by a factor of three.

00:05:32:06 - 00:05:35:12
On the A100 and H100 cards typically

00:05:35:12 - 00:05:37:01
available in compute clusters

00:05:37:01 - 00:05:38:09
the difference is smaller,

00:05:38:09 - 00:05:39:05
since those cards

00:05:39:05 - 00:05:41:00
have good 64-bit throughput,

00:05:41:00 - 00:05:42:23
but it's still a meaningful saving.

00:05:42:23 - 00:05:44:12
The challenge with 32-bit floating

00:05:44:12 - 00:05:46:11
point is precision.

00:05:46:11 - 00:05:48:02
A 32-bit float has around

00:05:48:02 - 00:05:49:06
7 significant

00:05:49:06 - 00:05:51:05
decimal digits of precision.

00:05:51:05 - 00:05:52:07
If your light curve flux

00:05:52:07 - 00:05:54:14
values are stored as normalised flux -

00:05:54:14 - 00:05:56:14
where the out-of-transit level is 1.0,

00:05:56:14 - 00:05:59:04
and in-transit flux is slightly less -

00:05:59:04 - 00:06:00:11
then your precision is limited

00:06:00:11 - 00:06:02:08
to around 1 part per million

00:06:02:08 - 00:06:04:10
at the level of the baseline.

00:06:04:10 - 00:06:05:14
For PLATO targets

00:06:05:14 - 00:06:07:20
where transit depths can be 100 parts

00:06:07:20 - 00:06:09:04
per million or less,

00:06:09:04 - 00:06:10:08
that's right at the edge of

00:06:10:08 - 00:06:11:09
what's acceptable.

00:06:11:09 - 00:06:12:04
The solution is

00:06:12:04 - 00:06:14:03
a simple reparameterisation.

00:06:14:03 - 00:06:16:06
Instead of storing flux as normalised

00:06:16:06 - 00:06:17:22
values centred on 1.0,

00:06:17:22 - 00:06:19:09
CETRA works internally

00:06:19:09 - 00:06:21:11
with flux offsets from the baseline -

00:06:21:11 - 00:06:22:07
that is,

00:06:22:07 - 00:06:23:23
the value 0 represents

00:06:23:23 - 00:06:25:21
no deviation from baseline,

00:06:25:21 - 00:06:27:02
and a transit appears

00:06:27:02 - 00:06:28:24
as a positive excursion.

00:06:28:24 - 00:06:29:24
With this convention,

00:06:29:24 - 00:06:31:22
the floating point precision scales

00:06:31:22 - 00:06:33:15
with the size of the deviation

00:06:33:15 - 00:06:36:00

rather than the absolute flux level,

00:06:36:00 - 00:06:37:07
and the potential for precision

00:06:37:07 - 00:06:39:12
loss essentially disappears.

00:06:39:12 - 00:06:40:22
A similar reparameterisation

00:06:40:22 - 00:06:42:13
is applied to time.

00:06:42:13 - 00:06:43:12
CETRA works in time

00:06:43:12 - 00:06:45:10
relative to the start of the light curve

00:06:45:10 - 00:06:47:12
rather than in absolute timestamps,

00:06:47:12 - 00:06:48:11
avoiding precision

00:06:48:11 - 00:06:50:22
loss for long light curves.

00:06:50:22 - 00:06:52:19
These are small changes to make,

00:06:52:19 - 00:06:53:18
but they matter

00:06:53:18 - 00:06:54:22
and they're a good example

00:06:54:22 - 00:06:55:21
of the kind of careful

00:06:55:21 - 00:06:58:15
thinking that GPU programming requires.

00:06:58:15 - 00:07:00:02
With the resampled light curve

00:07:00:02 - 00:07:01:21
and transit model in place,

00:07:01:21 - 00:07:03:11
the linear search sweeps over

00:07:03:11 - 00:07:05:04
a two-dimensional grid of transit

00:07:05:04 - 00:07:07:06
reference times and durations.

00:07:07:06 - 00:07:08:18
For each combination,

00:07:08:18 - 00:07:09:17
it identifies

00:07:09:17 - 00:07:11:11
the corresponding transit window

00:07:11:11 - 00:07:13:12
in the resampled light curve -

00:07:13:12 - 00:07:15:00
using the cheap index

00:07:15:00 - 00:07:17:00
arithmetic described above -

00:07:17:00 - 00:07:17:17
and performs

00:07:17:17 - 00:07:19:16
two passes over the data points

00:07:19:16 - 00:07:21:04
within that window.

00:07:21:04 - 00:07:22:16
The first pass computes

00:07:22:16 - 00:07:23:06
the maximum

00:07:23:06 - 00:07:24:06
likelihood transit

00:07:24:06 - 00:07:25:24
depth implied by all points

00:07:25:24 - 00:07:27:10
within the window.

00:07:27:10 - 00:07:29:00
Recall that resampled points

00:07:29:00 - 00:07:29:24
with no underlying

00:07:29:24 - 00:07:32:09
observations carry infinite uncertainty

00:07:32:09 - 00:07:34:04
and therefore contribute nothing -

00:07:34:04 - 00:07:35:12
data gaps have been handled

00:07:35:12 - 00:07:36:15
automatically.

00:07:36:15 - 00:07:38:09
The second pass computes

00:07:38:09 - 00:07:39:19
the likelihood ratio -

00:07:39:19 - 00:07:41:00
the likelihood of the data

00:07:41:00 - 00:07:42:08

given the transit model

00:07:42:08 - 00:07:43:22
relative to the likelihood

00:07:43:22 - 00:07:46:06
given a constant flux model.

00:07:46:06 - 00:07:47:24
To understand why this is useful,

00:07:47:24 - 00:07:49:08
it helps to write down

00:07:49:08 - 00:07:51:04
what we mean by the likelihood.

00:07:51:04 - 00:07:52:01
For any model

00:07:52:01 - 00:07:54:19
while assuming Gaussian measurement errors,

00:07:54:19 - 00:07:56:03
the log-likelihood of the observed

00:07:56:03 - 00:07:57:13
data is shown here,

00:07:57:13 - 00:08:00:06
where fn is the observed flux,

00:08:00:06 - 00:08:02:10
mn is the model flux, and sigma n

00:08:02:10 - 00:08:04:10
is the measurement uncertainty

00:08:04:10 - 00:08:05:24
all at data point n.

00:08:05:24 - 00:08:07:12
For the no-transit model

00:08:07:12 - 00:08:10:13
mn is 0 everywhere - because CETRA

00:08:10:13 - 00:08:11:23
works in flux offsets

00:08:11:23 - 00:08:12:21
from baseline,

00:08:12:21 - 00:08:15:04
a detrended light curve with no transit

00:08:15:04 - 00:08:16:22
present is just 0.

00:08:16:22 - 00:08:18:03
For the transit model

00:08:18:03 - 00:08:19:08
mn is 0

00:08:19:08 - 00:08:20:21
outside the transit window

00:08:20:21 - 00:08:23:10
and follows the transit shape inside it.

00:08:23:10 - 00:08:25:07
Now consider the likelihood ratio -

00:08:25:07 - 00:08:27:05
the log-likelihood of the transit model

00:08:27:05 - 00:08:28:16
minus the log-likelihood of the

00:08:28:16 - 00:08:30:05
no-transit model.

00:08:30:05 - 00:08:31:16
Outside the transit window

00:08:31:16 - 00:08:33:19
mn is 0 in both models,

00:08:33:19 - 00:08:35:09
so those terms are identical

00:08:35:09 - 00:08:37:07
and cancel exactly.

00:08:37:07 - 00:08:39:14
The second term inside the sum,

00:08:39:14 - 00:08:40:08
which involves

00:08:40:08 - 00:08:43:09
only sigma n, also cancels everywhere.

00:08:43:09 - 00:08:45:15
Here is what remains.

00:08:47:03 - 00:08:48:19
Two things follow from this.

00:08:48:19 - 00:08:51:08
First, we only need to evaluate points

00:08:51:08 - 00:08:52:22
inside the transit window -

00:08:52:22 - 00:08:54:03
which is exactly the cheap

00:08:54:03 - 00:08:55:03
index arithmetic

00:08:55:03 - 00:08:57:22
that the resampling step makes possible.

00:08:57:22 - 00:08:58:15
Second,

00:08:58:15 - 00:09:00:14

the log term has vanished entirely,

00:09:00:14 - 00:09:01:05
which avoids

00:09:01:05 - 00:09:04:18
an expensive operation in the inner loop.

00:09:04:18 - 00:09:06:10
Both are direct consequences

00:09:06:10 - 00:09:07:09
of working with

00:09:07:09 - 00:09:08:11
a detrended light curve

00:09:08:11 - 00:09:10:14
in offset representation.

00:09:10:14 - 00:09:11:20
The output of the linear

00:09:11:20 - 00:09:13:07
search is three

00:09:13:07 - 00:09:14:16
two-dimensional arrays,

00:09:14:16 - 00:09:16:20
one each for depth, depth variance,

00:09:16:20 - 00:09:18:11
and likelihood ratio -

00:09:18:11 - 00:09:19:04
with transit

00:09:19:04 - 00:09:20:05
reference time along

00:09:20:05 - 00:09:23:08
one axis and duration along the other.

00:09:23:08 - 00:09:25:17
These are computed entirely on the GPU

00:09:25:17 - 00:09:27:08
and they can remain there;

00:09:27:08 - 00:09:28:19
they don't need to be transferred

00:09:28:19 - 00:09:31:07
back to the host machine at this stage.

00:09:31:07 - 00:09:32:21
Each element of these output arrays

00:09:32:21 - 00:09:34:24
is computed by a single thread block.

00:09:34:24 - 00:09:36:06
Within a thread block,

00:09:36:06 - 00:09:37:19
the data points in the transit

00:09:37:19 - 00:09:40:06
window are divided between threads,

00:09:40:06 - 00:09:40:23
each thread

00:09:40:23 - 00:09:42:13
accumulating a partial sum,

00:09:42:13 - 00:09:44:17
and then a parallel reduction -

00:09:44:17 - 00:09:45:10
where threads

00:09:45:10 - 00:09:46:15
cooperatively combine

00:09:46:15 - 00:09:47:14
their partial results

00:09:47:14 - 00:09:49:11
in a series of steps - produces

00:09:49:11 - 00:09:50:05
the final result.

00:09:50:05 - 00:09:51:22
The reduction uses shared memory,

00:09:51:22 - 00:09:53:05
making it fast.

00:09:53:05 - 00:09:54:12
The transit reference time

00:09:54:12 - 00:09:55:15
step size is fixed

00:09:55:15 - 00:09:57:17
at 1% of the shortest trial duration

00:09:57:17 - 00:09:59:02
by default.

00:09:59:02 - 00:10:01:12
In TLS the equivalent step size varies

00:10:01:12 - 00:10:02:20
with the trial duration,

00:10:02:20 - 00:10:04:06
which is an efficient choice

00:10:04:06 - 00:10:05:20
that can result in coarser

00:10:05:20 - 00:10:06:18
effective coverage

00:10:06:18 - 00:10:07:23

for long-period planets

00:10:07:23 - 00:10:09:24
with short transit durations.

00:10:09:24 - 00:10:12:07
Because CETRA searches in linear time

00:10:12:07 - 00:10:13:21
rather than phase space,

00:10:13:21 - 00:10:14:13
a uniform

00:10:14:13 - 00:10:15:21
step size is straightforward

00:10:15:21 - 00:10:16:23
to maintain across

00:10:16:23 - 00:10:18:09
the full parameter grid,

00:10:18:09 - 00:10:19:12
and this turns out to be

00:10:19:12 - 00:10:21:09
a modest sensitivity advantage

00:10:21:09 - 00:10:22:24
for exactly the long-period

00:10:22:24 - 00:10:25:08
signals PLATO cares most about.

00:10:25:08 - 00:10:26:22
The periodic search takes

00:10:26:22 - 00:10:28:15
the output of the linear search -

00:10:28:15 - 00:10:29:05
those three

00:10:29:05 - 00:10:31:12
2D arrays of depth, depth variance,

00:10:31:12 - 00:10:32:23
and likelihood ratio -

00:10:32:23 - 00:10:34:02
and phase-folds them

00:10:34:02 - 00:10:36:15
over a grid of trial periods.

00:10:36:15 - 00:10:38:02
For each trial period,

00:10:38:02 - 00:10:40:07
it identifies all the transit windows

00:10:40:07 - 00:10:41:15
in the linear search output

00:10:41:15 - 00:10:42:14
that would correspond

00:10:42:14 - 00:10:45:05
to transits of a planet with that period

00:10:45:05 - 00:10:47:08
starting at each possible reference

00:10:47:08 - 00:10:48:17
time and duration.

00:10:48:17 - 00:10:50:14
Because the linear search output

00:10:50:14 - 00:10:52:01
is on a regular grid,

00:10:52:01 - 00:10:53:05
identifying the relevant

00:10:53:05 - 00:10:55:16
elements is again simple arithmetic -

00:10:55:16 - 00:10:57:10
no sorting is required.

00:10:57:10 - 00:10:58:13
It then computes the

00:10:58:13 - 00:10:59:23
inverse-variance weighted

00:10:59:23 - 00:11:02:17
mean depth across those windows -

00:11:02:17 - 00:11:04:00
enforcing the physically

00:11:04:00 - 00:11:05:02
motivated assumption

00:11:05:02 - 00:11:06:22
that all transits of the same planet

00:11:06:22 - 00:11:08:24
should have the same depth -

00:11:08:24 - 00:11:09:19
and combines

00:11:09:19 - 00:11:10:24
their likelihood ratios

00:11:10:24 - 00:11:11:17
into a joint

00:11:11:17 - 00:11:12:16
likelihood ratio

00:11:12:16 - 00:11:15:02
for the periodic signal as a whole.

00:11:15:02 - 00:11:16:03

The period and duration

00:11:16:03 - 00:11:16:20
grids used

00:11:16:20 - 00:11:18:07
are the same astrophysically

00:11:18:07 - 00:11:20:05
motivated grids as TLS,

00:11:20:05 - 00:11:22:03
which ensures that the search

00:11:22:03 - 00:11:23:12
is neither over-sampled

00:11:23:12 - 00:11:24:07
and wasting

00:11:24:07 - 00:11:26:01
compute time, nor under-sampled

00:11:26:01 - 00:11:27:16
and missing signals.

00:11:27:16 - 00:11:28:13
One important

00:11:28:13 - 00:11:30:16
implementation detail concerns

00:11:30:16 - 00:11:33:04
what gets returned from the GPU.

00:11:33:04 - 00:11:34:00
In principle,

00:11:34:00 - 00:11:35:17
the periodic search produces

00:11:35:17 - 00:11:37:12
a full set of depth,

00:11:37:12 - 00:11:38:12
depth variance,

00:11:38:12 - 00:11:40:00
and likelihood ratio arrays

00:11:40:00 - 00:11:41:18
for every trial period.

00:11:41:18 - 00:11:42:10
But copying

00:11:42:10 - 00:11:43:13
all that back to the host

00:11:43:13 - 00:11:46:08
machine would be a major bottleneck.

00:11:46:08 - 00:11:47:04
In practice,

00:11:47:04 - 00:11:48:09
for most purposes,

00:11:48:09 - 00:11:50:00
you only care about the maximum

00:11:50:00 - 00:11:52:03
likelihood result for each period.

00:11:52:03 - 00:11:53:13
So the periodic search

00:11:53:13 - 00:11:54:20
kernel incorporates

00:11:54:20 - 00:11:56:19
an on-device reduction:

00:11:56:19 - 00:11:59:00
it finds the maximum likelihood element

00:11:59:00 - 00:12:00:13
within the output arrays

00:12:00:13 - 00:12:02:01
for each trial period,

00:12:02:01 - 00:12:02:17
and returns

00:12:02:17 - 00:12:04:12
only that and the corresponding

00:12:04:12 - 00:12:06:03
model parameters.

00:12:06:03 - 00:12:09:01
The full arrays never leave the GPU.

00:12:09:01 - 00:12:11:09
The outcome is a periodogram -

00:12:11:09 - 00:12:12:21
for each trial period,

00:12:12:21 - 00:12:14:17
the maximum likelihood ratio

00:12:14:17 - 00:12:17:09
and the corresponding transit parameters.

00:12:17:09 - 00:12:18:24
A peak in this periodogram

00:12:18:24 - 00:12:19:14
at the true

00:12:19:14 - 00:12:21:14
period of an injected signal -

00:12:21:14 - 00:12:23:14
rising clearly above the noise floor

00:12:23:14 - 00:12:25:11

and the aliases at half and double

00:12:25:11 - 00:12:27:08
the period - is the signature

00:12:27:08 - 00:12:29:09
of a detected planet.